JOSÉ ALBERTO GUTIÉRREZ ROBLES
MIGUEL ÁNGEL OLMOS GÓMEZ
JUAN MARTÍN CASILLAS GONZÁLEZ

Análisis NUMÉRICO

Mc Graw Hill

PDE Portada Diseñada por Estudiantes
www.FreeLibros.me

www.FreeLibros.me

Análisis numérico

www.FreeLibros.me

www.FreeLibros.me

Análisis numérico

José Alberto Gutiérrez Robles
Miguel Ángel Olmos Gómez
Juan Martín Casillas González
Universidad de Guadalajara

Revisión técnica

José Job Flores Godoy
Universidad Iberoamericana
Ciudad de México

MÉXICO • BOGOTÁ • BUENOS AIRES • CARACAS • GUATEMALA • MADRID • NUEVA YORK
SAN JUAN • SANTIAGO • SÃO PAULO • AUCKLAND • LONDRES • MILÁN • MONTREAL
NUEVA DELHI • SAN FRANCISCO • SINGAPUR • ST. LOUIS • SIDNEY • TORONTO

www.FreeLibros.me

Director Higher Education: Miguel Ángel Toledo C.
Editor sponsor: Pablo E. Roig Vázquez
Coordinadora editorial: Marcela I. Rocha M.
Editora de desarrollo: Ana Laura Delgado R.
Supervisor de producción: Zeferino García G.

ANÁLISIS NUMÉRICO

Prohibida la reproducción total o parcial de esta obra,
por cualquier medio, sin la autorización escrita del editor.

McGraw Hill Educación

DERECHOS RESERVADOS © 2010, respecto a la primera edición por
McGRAW-HILL/INTERAMERICANA EDITORES, S.A. DE C.V.
*A Subsidiary of **The McGraw-Hill** Companies, Inc.*
Edificio Punta Santa Fe
Prolongación Paseo de la Reforma 1015, Torre A
Piso 17, Colonia Desarrollo Santa Fe
Delgación Álvaro Obregón
C.P. 01376, México, D.F.
Miembro de la Cámara Nacional de la Industria Editorial Mexicana, Reg. Núm. 736

ISBN: 978-607-15-0316-9

1234567890 109876543210

Impreso en México *Printed in Mexico*

The McGraw-Hill Companies

www.FreeLibros.me

Contenido

www.FreeLibros.me

www.FreeLibros.me

www.FreeLibros.me

www.FreeLibros.me

Acerca de los autores

Dr. José Alberto Gutiérrez Robles es graduado de la licenciatura y la maestría en la Universidad de Guadalajara, en 1995 y 1998, respectivamente. Doctorado en Ciencias de la Ingeniería Eléctrica en Cinvestav, Unidad Guadalajara, en 2002, con una estancia de un año en la Universidad de Bolonia, Italia. Ha participado en siete publicaciones en revistas arbitradas y más de veinte artículos de conferencia internacional. El doctor Gutiérrez tiene como área de desarrollo los transitorios electromagnéticos producidos por descargas atmosféricas y la implementación numérica de esquemas de solución para sistemas de ecuaciones diferenciales. Actualmente es profesor de tiempo completo en el Departamento de Matemáticas de la Universidad de Guadalajara.

Dr. Miguel Ángel Olmos Gómez estudió la licenciatura en Matemáticas en la Universidad de Guadalajara del año 1980 al 1985, y el doctorado en Washington State University durante 1990 y 1994. Ha trabajado en la Universidad del Valle de Atemajac, el Instituto Tecnológico de Estudios Superiores de Monterrey, campus Guadalajara, la Universidad del Ejército y Fuerza Aérea Mexicana y la Universidad de Guadalajara. Ha dirigido 16 tesis de licenciatura en Matemáticas y seis de Maestría en Matemática Aplicada. Cuenta con nueve publicaciones en revistas arbitradas y ha impartido más de 50 conferencias en diferentes foros nacionales e internacionales.

M. en C. Juan Martín Casillas González estudió la licenciatura en Fisicomatemáticas en la Universidad de Guadalajara entre 1989 y 1994 y la maestría en Ciencias en Telecomunicaciones, en Cinvestav, Unidad Guadalajara de 1997 a 1999. Actualmente es estudiante del Doctorado en Ciencias y Tecnología, con sede en Lagos de Moreno, Jalisco, perteneciente a la Universidad de Guadalajara. Su trabajo de tesis está orientado a establecer esquemas numéricos no estándar para la solución de ecuaciones diferenciales parciales no lineales. Como profesor está orientado al área docente, lo que lo llevó a ser designado coordinador académico del programa de estudios de Licenciatura en Matemáticas en la Universidad de Guadalajara en el periodo 2007-2010, así como jefe de la academia de Matemáticas Aplicadas.

www.FreeLibros.me

www.FreeLibros.me

Prefacio

El desarrollo de los sistemas de cómputo y la capacidad de realizar miles o millones de operaciones por segundo, le da a la implementación numérica el marco adecuado para su desarrollo. Los sistemas de ecuaciones de gran tamaño, cuya solución numérica era impensable, se vuelven problemas cotidianos. Así, el desarrollo o análisis numérico de las soluciones de la representación matemática de fenómenos físicos de gran complejidad toma enorme relevancia. El esfuerzo entonces se centra en encontrar nuevas formas y metodologías de solución numérica para todo tipo de funciones o sistemas de funciones, sean lineales o no lineales. Es en esta área donde se centran los temas expuestos en este libro; por supuesto se hace énfasis del hecho de que hay más de una forma de resolver un mismo problema, y se dan las ventajas y desventajas de los diferentes métodos numéricos presentados, de igual forma se presenta el marco teórico que fundamenta cada método.

Primero, se introducen los conceptos fundamentales del cálculo computacional y del cálculo diferencial que son de particular importancia en el resto del libro. Se aborda de manera amplia la solución de ecuaciones no lineales de una sola variable, haciendo una diferenciación clara y precisa de los métodos adecuados para resolver las de tipo polinomial; para este tipo de ecuaciones no lineales se pone particular énfasis en la aritmética empleada así como del tipo, alcance y limitante de cada método, de la misma forma se hace una revisión de los métodos para sistemas de ecuaciones no lineales.

Respecto de los sistemas de ecuaciones lineales, se consideran las formas de solución más conocidas; en forma adicional se hace el análisis de casos especiales, facilitando la matemática y la metodología de solución de estos casos, como los sistemas subdeterminados y sobredeterminados. Se describe de manera amplia el concepto de interpolación y los métodos más utilizados, en forma adicional se presenta de manera original lo referente a ajuste de curvas sea utilizando el concepto de mínimos cuadrados, como la serie de Fourier o los polinomios ortogonales. A partir de polinomios interpoladores se obtienen metodológicamente los métodos de derivación e integración numérica; igualmente se explica de manera clara y metodológica la forma de obtener y generar los métodos de solución de ecuaciones y sistemas de ecuaciones diferenciales ordinarias. Se presenta el concepto de diagonalización y las formas canónicas de Jordan, como el antecedente de la solución analítica de un sistema de ecuaciones diferenciales ordinarias, se concluye este tema con una revisión de los métodos de cálculo de valores y vectores propios. Por último, se aborda la solución de ecuaciones diferenciales parciales presentando algunas ideas básicas de su implementación con diferencias finitas.

Por supuesto, tanto los ejercicios cuya solución se presenta en el libro como aquellos que se proponen, son de gran importancia para los estudiantes, por esta razón en cada

www.FreeLibros.me

capítulo se introducen ejercicios que refuerzan los conceptos e ideas fundamentales de cada tema; también se proponen al final de cada capítulo una serie de problemas cuya solución se deja a los usuarios de este material. Es de suma relevancia entender que este material, aunque expuesto de manera original, no es único, es decir, para casi todos los temas existen otros textos que lo tratan de manera diferente o en algunos casos similar, por lo que se incluye una referencia bibliográfica adecuada que ayuda a los estudiantes a encontrar formas adicionales de abordar los temas.

www.FreeLibros.me

Capítulo 1

Cálculo computacional

1.1 Introducción

Antes de iniciar el estudio de los métodos computacionales, es necesario analizar por qué se ha incrementado tanto el uso de las computadoras para el cálculo científico en las últimas décadas. Aunque nuestro cerebro es capaz de retener y procesar una gran cantidad de información, pocos serían capaces de hacer, con precisión y suficiente rapidez, una operación tan simple como 1.23454×6.76895. Sin embargo, una computadora es capaz de hacer miles o millones de estas operaciones por segundo.

Sin embargo, en el caso específico de una computadora existen dos tipos de errores en los cálculos numéricos. El primero, llamado *error de truncamiento*, se debe a las aproximaciones utilizadas en las fórmulas matemáticas de los modelos. El segundo, llamado *error de redondeo*, se asocia al número finito de dígitos con los que se representan los números en una computadora. Para el análisis de los errores de truncamiento, normalmente se utiliza la serie de Taylor, y para los errores de redondeo, se analiza la forma de almacenamiento de datos de una computadora, así como la forma de procesarlos, es decir, de hacer las operaciones [Burden *et al.*, 2002], [Chapra *et al.*, 2007], [Cheney *et al.*, 2008].

1.2 Preliminares matemáticos

En esta sección se establecen los preliminares matemáticos necesarios para el análisis numérico. Esta revisión no es exhaustiva; sólo pretende hacer un recuento de los resultados fundamentales necesarios. Se considera que el lector ha estudiado y conoce las definiciones de límite de una función, continuidad de funciones y convergencia de sucesiones. Estos temas se pueden consultar en cualquier libro de cálculo elemental [Allen *et al.,* 1998], [Burden *et al.*, 2002]. En forma inicial se establece la siguiente notación:

Notación

- $C(X)$ denota el conjunto de todas las funciones reales continuas sobre el conjunto X.
- $C[a, b]$ denota el conjunto de todas las funciones reales continuas definidas sobre el intervalo $[a, b]$.
- El símbolo $f \in C[a, b]$ significa que la función $f : [a, b] \to \Re$ es continua en el intervalo $[a, b]$.
- El conjunto de todas las funciones con n derivadas continuas en X se denota por $C^n(X)$.

www.FreeLibros.me

- Si $f \in C^n(X)$, la función f se dice que es de clase n en X y significa que hasta la n derivada de f existe y es continua en X.

Teorema 1.1 (Teorema del valor intermedio) Si $f \in C[a, b]$ y K es un número cualquiera entre $f(a)$ y $f(b)$, entonces existe $c \in [a, b]$ tal que $f(c) = K$.

Este teorema afirma que si una función continua cambia de signo en los extremos de un intervalo debe tener un cero en el intervalo.

Otro resultado acerca de funciones continuas definidas sobre intervalos cerrados es el siguiente teorema.

Teorema 1.2 (Teorema de los valores extremos) Si $f \in C[a, b]$, entonces existen $x_1, x_2 \in [a, b]$ tales que $f(x_1) \leq f(x) \leq f(x_2)$ para toda $x \in [a, b]$.

Los siguientes teoremas están relacionados con la derivabilidad de las funciones. La importancia radica en su empleo para determinar los errores en los diferentes métodos numéricos que se desarrollen en esta obra. Se inicia con:

Teorema 1.3 (Teorema de Rolle) Sea $f \in C[a, b]$, y derivable en (a, b). Si $f(a) = f(b)$ entonces existe al menos un número $c \in (a, b)$ tal que $f'(c) = 0$. Este teorema se puede generalizar como

Teorema 1.4 (Teorema de Rolle generalizado) Si $f \in C^n[a, b]$ y existen $n+1$ puntos distintos $x_0, x_1, \ldots, x_n \in [a, b]$ tal que $f(x_0) = f(x_1) = \cdots = f(x_n)$ entonces existe $c \in (a, b)$ tal que $f^{(n)}(c) = 0$

Como corolario al Teorema de Rolle se tiene el siguiente teorema:

Teorema 1.5 (Teorema del valor medio) Si $f \in C[a, b]$, y f es derivable en (a, b), existe un número $c \in (a, b)$ tal que $f(b) - f(a) = f'(c)(b - a)$.

También se puede establecer el teorema de valores extremos para funciones derivables como:

Teorema 1.6 (Teorema de los valores extremos) Si $f \in C[a, b]$, y f es derivable en (a, b), entonces existen $x_1, x_2 \in [a, b]$ tales que $f(x_1) \leq f(x) \leq f(x_2)$ para toda $x \in [a, b]$. Si además se tiene que $f'(x_1) = 0$, $f'(x_2) = 0$, a $f(x_1)$ y $f(x_2)$ se les llama valores extremos de f en $[a, b]$.

En la siguiente definición se extiende el concepto de continuidad de una función como:

Definición 1.1 (Condición de Lipschitz) Se dice que $f: [a, b] \to \Re$ satisface una condición de Lipschitz con constante de Lipschitz L en $[a, b]$ si, para cada $x, y \in [a, b]$ se tiene que $|f(y) - f(x)| \leq L|y - x|$.

www.FreeLibros.me

Por lo anterior se tienen los siguientes resultados:

Teorema 1.7 Si $f:[a,b]\to\Re$ satisface una condición de Lipschitz con constante de Lipschitz L en $[a,b]$ entonces $f\in C[a,b]$. Si además se tiene que f es derivable en (a,b) entonces f satisface la condición de Lipschitz con constante de Lipschitz $L=\sup_{x\in[a,b]}|f'(x)|$.

Por último, se proporcionan dos resultados importantes relacionados con la integral definida:

Teorema 1.8 (Teorema del valor medio ponderado para integrales) Si $f\in C[a,b]$, y g es integrable en $[a,b]$ y g no cambia de signo en el intervalo $[a,b]$. Existe $c\in(a,b)$ tal que

$$\int_a^b f(x)g(x)dx = f(c)\int_a^b g(x)dx$$

El siguiente resultado es un caso particular del teorema anterior cuando $g(x)=1,\ x\in[a,b]$.

Teorema 1.9 (Teorema del valor medio para integrales) Suponiendo que $f\in C[a,b]$, existe $c\in(a,b)$ tal que

$$f(c)=\frac{1}{b-a}\int_a^b f(x)dx$$

1.3 Series de Taylor

Las series que se presentan en las aplicaciones son las que se estudian formalmente en el cálculo elemental con el nombre de *series de Taylor*. Su estudio sistemático se inicia con el teorema de Taylor, el cual se puede expresar de la siguiente manera:

Teorema 1.10 Si $f(z)$ es analítica en toda la región limitada por una curva simple cerrada C y si z y a son puntos interiores de C, entonces

$$f(z)=f(a)+f'(a)(z-a)+f''(a)\frac{(z-a)^2}{2!}+\cdots+f^{n-1}(a)\frac{(z-a)^{n-1}}{(n-1)!}+R_n$$

Donde $R_n=\dfrac{(z-a)^n}{2\pi i}\displaystyle\int_C\frac{f(t)dt}{(t-a)^n(t-z)}$

Demostración Primero se observa que después de sumar y restar a en el denominador del integrando, la fórmula de la integral de Cauchy se puede escribir en la forma

$$f(z)=\frac{1}{2\pi i}\int_C\frac{f(t)dt}{t-z}=\frac{1}{2\pi i}\int_C\frac{f(t)}{t-a}\left(\frac{1}{1-\dfrac{z-a}{t-a}}\right)dt$$

entonces aplicando la identidad $\dfrac{1}{1-u}=1+u+u^2+u^3+\cdots+u^{n-1}+\dfrac{u^n}{1-u}$ al factor entre paréntesis se tiene que

www.FreeLibros.me

$$f(z)=\frac{1}{2\pi i}\int_C \frac{f(t)}{t-a}\left[1+\frac{z-a}{t-a}+\left(\frac{z-a}{t-a}\right)^2+\cdots+\left(\frac{z-a}{t-a}\right)^{n-1}+\frac{\frac{(z-a)^n}{(t-a)^n}}{1-\frac{z-a}{t-a}}\right]dt$$

o bien,

$$f(z)=\frac{1}{2\pi i}\int_C \frac{f(t)dt}{t-a}+\frac{z-a}{2\pi i}\int_C \frac{f(t)dt}{(t-a)^2}+\cdots+\frac{(z-a)^{n-1}}{2\pi i}\int_C \frac{f(t)dt}{(t-a)^n}+\frac{(z-a)^n}{2\pi i}\int_C \frac{f(t)dt}{(t-a)^n(t-z)}$$

A partir de las generalizaciones de la fórmula de la integral de Cauchy es evidente que, excepto por los factoriales necesarios, las primeras n integrales de la última expresión son precisamente las derivadas correspondientes de $f(z)$, evaluadas en el punto $z=a$. Por tanto,

$$f(z)=f(a)+f'(a)(z-a)+f''(a)\frac{(z-a)^2}{2!}+\cdots+f^{n-1}(a)\frac{(z-a)^{n-1}}{(n-1)!}+\frac{(z-a)^n}{2\pi i}\int_C \frac{f(t)dt}{(t-a)^n(t-z)}$$

lo cual demuestra el teorema.

Serie de Taylor

Por serie de Taylor se entiende el desarrollo infinito que sugiere el último teorema, es decir,

$$f(z)\approx f(a)+f'(a)(z-a)+f''(a)\frac{(z-a)^2}{2!}+\cdots+f^{n-1}(a)\frac{(z-a)^{n-1}}{(n-1)!}+\cdots$$

Para demostrar que esta serie converge realmente a $f(z)$ hay que demostrar que el valor absoluto de la diferencia entre $f(z)$ y la suma de los n primeros términos de la serie tiende a cero cuando n tiende a infinito. Por el teorema de Taylor es evidente que esta diferencia es:

$$R_n=\frac{(z-a)^n}{2\pi i}\int_C \frac{f(t)dt}{(t-a)^n(t-z)}$$

en consecuencia, se deben determinar los valores de z para los cuales el valor absoluto de esta integral tiende a cero cuando n tiende a infinito.

Para ello, sean C_1 y C_2 dos círculos de radios r_1 y r_2 cuyos centros están en a y situados completamente en el interior de C (véase figura 1.1). Como $f(z)$ es analítica en todo el interior de C, el integrando completo de $R_n(z)$ es analítico en la región entre C_1 y C_2 siempre que z, como a, esté en el interior de C_2. Bajo estas condiciones la integral alrededor de C_1 se puede reemplazar por la integral en torno a C_2. Si, además, z está en el interior de C_1, entonces, para todos los valores de t sobre C_2, se tiene:

$$|t-a|=r_2 \text{ y } |z-a|<r_1$$

Por tanto

$$|t-z|>r_2-r_1$$

Así, se tiene que

$$|f(t)|\le M$$

donde M es el máximo de $|f(z)|$ sobre C_2.

www.FreeLibros.me

El valor absoluto de la diferencia, sobrestimando los factores del numerador y subestimando los denominadores es:

$$|R_n(z)| = \left| \frac{(z-a)^n}{2\pi i} \int_C \frac{f(t)dt}{(t-a)^n (t-z)} \right|$$

De la igualdad anterior, la teoría de valor absoluto establece que:

$$|R_n(z)| \leq \frac{|z-a|^n}{|2\pi i|} \int_C \frac{|f(t)||dt|}{|t-a|^n |t-z|}$$

Utilizando las definiciones anteriores se llega a:

$$|R_n(z)| < \frac{r_1^n}{2\pi} \int_C \frac{M|dt|}{r_2^n (r_2 - r_1)}$$

Resolviendo la integral cerrada en r_2 se obtiene

$$|R_n(z)| < \frac{r_1^n M}{2\pi r_2^n (r_2 - r_1)} 2\pi r_2$$

Simplificando términos finalmente se tiene que el valor absoluto de la diferencia es:

$$|R_n(z)| < M \left(\frac{r_1}{r_2} \right)^n \frac{r_2}{r_2 - r_1}$$

Como $0 < r_1 < r_2$, la fracción $(r_1/r_2)^n$ tiende a cero cuando n tiende a infinito; por tanto, el límite de $|R_n(z)|$ es cero; así se, demuestra, el teorema.

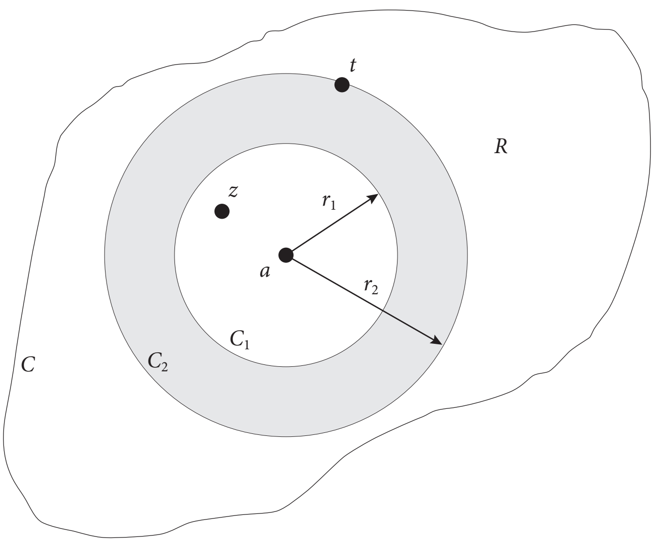


Figura 1.1 Los círculos C_1 y C_2 utilizados en la demostración de la convergencia de la serie de Taylor.

www.FreeLibros.me

Lo anterior se puede reescribir definiendo $h = z - a$ como,

Teorema 1.11 Si $f(z)$ es analítica en toda la región limitada por una curva simple cerrada C y si z y a están en el interior de C, entonces

$$f(z) = f(a) + hf'(a) + \frac{h^2}{2!} f''(a) + \cdots + \frac{h^{n-1}}{(n-1)!} f^{n-1}(a) + R_n$$

Donde $R_n = \dfrac{h^n}{2\pi i} \displaystyle\int_C \frac{f(t)dt}{(t-a)^n (t-z)}$

Por tanto se concluye que,

Teorema 1.12 La serie de Taylor

$$f(z) = f(a) + f'(a)(z-a) + f''(a)\frac{(z-a)^2}{2!} + f'''(a)\frac{(z-a)^3}{3!} + \cdots$$

es una representación válida de $f(z)$ en todos los puntos en el interior de cualquier círculo que tenga su centro a y dentro del cual $f(z)$ sea analítica.

Cuando se consideran funciones reales se tiene el teorema siguiente.

Teorema 1.13 Si $f \in C^{n+1}[a,b]$ y $x_0 \in (a,b)$. Se tiene que para cualquier $x \in [a,b]$ existe ξ entre x_0 y x tal que

$$f(x) = f(x_0) + (x-x_0) f'(x_0) + \frac{(x-x_0)^2}{2!} f''(x_0) + \cdots + \frac{(x-x_0)^{n-1}}{(n-1)!} f^{n-1}(x_0) + R_n$$

donde $R_n = \dfrac{(x-x_0)^n}{n!} f^n(\xi)$.

n-ésimo polinomio de Taylor

Definimos el n-ésimo polinomio de Taylor como el polinomio de grado n obtenido de la serie de Taylor, al considerar sólo los términos hasta el grado n. Esto es

$$p_n(z) = f(a) + (z-a) f'(a) + \frac{(z-a)^2}{2!} f''(a) + \cdots + \frac{(z-a)^n}{n!} f^n(z-a)$$

Se tiene además que

$$f(z) = p_n(z) + R_{n+1}$$

Teorema 1.14 Si la serie de potencias $c_0 + c_1(z-a) + c_2(z-a)^2 + c_3(z-a)^3 + \cdots$ converge para $z = z_1$, entonces converge absolutamente para todos los valores de z tales que $|z-a| < |z_1 - a|$ y converge de manera uniforme para todos los valores de z tales que $|z-a| \le r < |z_1 - a|$.

www.FreeLibros.me

EJEMPLO 1.1

Para determinar el cuarto polinomio de Taylor correspondiente a $f(z) = \text{sen}(z) + \cos(z)$ en $z = 0$ se calcula hasta la cuarta derivada de $f(z)$

$$f'(z) = \cos(z) - \text{sen}(z), f''(z) = -\text{sen}(z) - \cos(z)$$
$$f'''(z) = -\cos(z) + \text{sen}(z), f^{iv}(z) = \text{sen}(z) + \cos(z)$$

Evaluando las derivadas en $z = 0$ se tiene que,

$$f(0) = 1, f'(0) = 1, f''(0) = -1, f'''(0) = -1, f^{iv}(0) = 1$$

Por lo que

$$p_4(z) = 1 + z - \frac{1}{2!}z^2 - \frac{1}{3!}z^3 + \frac{1}{4!}z^4$$

El residuo está dado por

$$R_5 = \frac{z^5}{5!}f^5(\xi) = \frac{z^5}{120}(\cos(\xi) - \text{sen}(\xi)) \text{ donde } \xi \text{ está entre 0 y } z.$$

Se tiene además que

$$|R_5| = \left|\frac{z^5}{120}(\cos(\xi) - \text{sen}(\xi))\right| \leq \frac{\sqrt{2}}{120}|z|^5$$

De esto se sigue que la aproximación del polinomio a la función es mejor si $|z| \ll 1$.

EJEMPLO 1.2

Dada la función $f(z) = \ln(z+1)$, $z \geq 0$. Se tiene

$$f'(z) = \frac{1}{z+1}, f''(z) = -\frac{1}{(z+1)^2}, f'''(z) = \frac{2!}{(z+1)^3}, f^{iv}(z) = -\frac{3!}{(z+1)^4}, f^{v}(z) = \frac{4!}{(z+1)^5}, \ldots$$

Evaluando las derivadas en $z = 0$ se obtiene

$$f(0) = 0, f'(0) = 1, f''(0) = -1, f'''(0) = 2!, f^{iv}(0) = -3!, f^{v}(0) = 4!, \ldots, f^{n}(0) = (n-1)!$$

Por lo que

$$p_n(z) = z - \frac{1}{2}z^2 + \frac{1}{3}z^3 - \frac{1}{4}z^4 + \frac{1}{5}z^5 + \cdots + \frac{1}{n}z^n$$

El residuo está dado por

$$R_n = \frac{z^n}{n!}f^n(\xi) = \frac{z^n}{n}\frac{(-1)^{n+1}}{(\xi+1)^n} \text{ donde } \xi \text{ está entre 0 y } z.$$

Además el residuo satisface

$$|R_n| \leq \left|\frac{z^n}{n}\right|$$

Si se denota por Γ el círculo más grande con centro en $z = a$ y dentro del cual $f(z)$ es analítica en todo punto, el teorema 1.12 garantiza que la serie de Taylor converge a $f(z)$ en todos los puntos en el interior de Γ. Sin embargo, no proporciona información alguna acerca del comportamiento de la serie fuera de Γ. En realidad, la serie de Taylor sólo converge a $f(z)$ dentro de Γ y, posiblemente sobre el círculo Γ y diverge, o bien, converge a un límite diferente de $f(z)$ en todo punto fuera de Γ.

www.FreeLibros.me

Demostración Como la serie dada converge cuando $z = z_1$, se deduce que los términos de la serie son acotados para este valor de z. En otras palabras, que existe una constante positiva M tal que

$$\left|c_n(z_1-a)^n\right| = |c_n| \cdot |z_1-a|^n \leq M \qquad \text{para } n = 0, 1, 2, \ldots,$$

Sea ahora z_0 cualquier valor de z tal que

$$|z_0-a| < |z_1-a|$$

es decir, sea z_0 cualquier punto más cercano a a que z_1. Entonces, para el término general de la serie cuando $z = z_0$, se tiene que

$$\left|c_n(z_0-a)^n\right| = |c_n| \cdot |z_0-a|^n = |c_n| \cdot |z_1-a|^n \cdot \left|\frac{z_0-a}{z_1-a}\right|^n \leq M\left|\frac{z_0-a}{z_1-a}\right|$$

Si se hace

$$\left|\frac{z_0-a}{z_1-a}\right|^n = k \tag{1.1}$$

donde k es, desde luego, menor que 1, esto muestra que los valores absolutos de los términos de la serie

$$c_0 + c_1(z-a) + c_2(z-a)^2 + c_3(z-a)^3 + \cdots \tag{1.2}$$

son superados, respectivamente, por los términos de la serie de constantes positivas

$$M + Mk + Mk^2 + Mk^3 + \cdots \tag{1.3}$$

Ésta es una serie geométrica cuya razón común k es numéricamente menor que 1. Por tanto, converge y proporciona una prueba de comparación que establece la convergencia absoluta de la serie dada por la ecuación (1.2). Sin embargo, la serie de la ecuación (1.3) no proporciona una serie de prueba que se pueda utilizar al aplicar la prueba M de Weierstrass a la serie de la ecuación (1.2), ya que resulta claro de la definición de k contenida en la ecuación (1.1) que los términos de la ecuación (1.3) dependen de z_0. Por tanto, para valores de z_0 tales que

$$|z_0-a| \leq r < |z_1-a| \tag{1.4}$$

se tiene por ejemplo que $k = \left|\frac{z_0-a}{z_1-a}\right| \leq \frac{r}{|z_1-a|} = \lambda$, y λ es, evidentemente, una constante menor que 1 independiente del punto general z_0. Luego, para todos los valores de z_0 que satisfagan la condición dada por la ecuación (1.4), la serie dada por la ecuación (1.2) es superada término a término por la serie geométrica convergente de constantes positivas $M + M\lambda + M\lambda^2 + M\lambda^3 + \cdots$ y, por tanto, la serie de la ecuación (1.2) converge uniformemente.

Por último, como cada término $c_n(z-a)^n$ de la serie dada es una función analítica y como cualquier punto en el interior del círculo $|z-a| = |z_1-a|$ se puede incluir dentro del círculo de la forma $|z-a| = r < |z_1-a$, entonces se establece que, dentro del círculo $|z-a| = |z_1-a|$, la función a la que converge la serie es analítica.

Sea α el punto singular de $f(z)$ más cercano al centro del desarrollo, $z = a$, y suponiendo que la serie de Taylor de $f(z)$ converge para algún valor de $z = z_1$, más alejado de a que α. La serie debe converger en todos los puntos que estén más cerca de a de lo que está z_1 y, es más, la suma a la cual converge debe ser analítica en todos los puntos. Por tanto, la serie converge a la función que es analítica en α, lo que, en apariencia, contradice la suposición de que α es un punto en el que $f(z)$ no es analítica. De donde, no se debe suponer que la serie converge en puntos más alejados de a que la distancia $|\alpha-a|$, o bien, se debe aceptar el hecho de que la función a la cual converge la serie, la vecindad de α, es diferente de $f(z)$. La situación usual es que la serie sea divergente para todos los valores de z tales que $|z-a| > |\alpha-a|$. Sin embargo, es posible que para valores de z tales que $|z-a| > |\alpha-a|$ la serie de Taylor de $f(z)$ converja a una función analítica diferente de $f(z)$.

www.FreeLibros.me

Teorema 1.15 Es imposible que la serie de Taylor de una función $f(z)$ converja a $f(z)$ fuera del círculo cuyo centro es el punto de desarrollo $z = a$ y cuyo radio es la distancia de a a la singularidad más cercana de $f(z)$.

Demostración El círculo más grande que se puede trazar alrededor del punto de desarrollo $z = a$, tal que la serie de Taylor de $f(z)$ converja en todo punto de su interior, se llama *círculo de convergencia* de la serie. En general, el radio de convergencia de la serie de Taylor de $f(z)$ alrededor del punto $z = a$ es igual a la distancia de a, a la singularidad más cercana de $f(z)$ y, en todo caso, es al menos tan grande como esta distancia. En particular, si el *punto singular* más cercano α tiene la propiedad de que $f(z)$ se hace infinita conforme z tiende a α, entonces el radio de convergencia es igual a $|\alpha - a|$. Por supuesto, todo este análisis se aplica sin cambio al caso en el que $a = 0$, el cual se conoce comúnmente como *serie de Maclaurin*.

La noción del círculo de convergencia a menudo es útil para determinar el intervalo de convergencia de una serie resultante del desarrollo de una función de una variable real. Con el fin de ilustrar lo anterior, se considera

$$f(z) = \frac{1}{1+z^2} = 1 - z^2 + z^4 - z^6 + \cdots$$

Ésta convergerá en todo el círculo de mayor radio alrededor del origen en el que $f(z)$ sea analítica. Ahora bien, por inspección, $f(z)$ se hace infinita conforme z se aproxima a $\pm i$ y aun cuando es posible que sólo se tenga interés en los valores reales de z, estas singularidades en el plano complejo establecen un límite insalvable en el intervalo de convergencia sobre el eje x. De hecho, se puede tener convergencia alrededor de $x = a$ sobre el eje real, únicamente sobre el diámetro horizontal del círculo de convergencia alrededor del punto $z = a$ en el plano complejo. Como una aplicación del desarrollo de Taylor se establece un resultado importante conocido como teorema de Liouville.

Teorema 1.16 (de Liouville) Si $f(z)$ es acotada y analítica para todo el valor de z, entonces $f(z)$ es constante.

Demostración Se observa que como $f(z)$ es analítica en todo punto, posee un desarrollo en serie de potencias alrededor del origen

$$f(z) = f(0) + f'(0)z + \cdots + \frac{f^n(0)}{n!}z^n$$

el cual converge y la representa para todo valor de z. Ahora bien, si C es un círculo arbitrario con centro en el origen, resulta, por la desigualdad de Cauchy, que

$$|f^n(0)| \le \frac{n!M_C}{r^n}$$

donde r es el radio de C y M_C es el valor máximo de $|f(x)|$ sobre C. Luego, para el coeficiente de z^n en el desarrollo de $f(z)$, se tiene que:

$$\left|\frac{f^n(0)}{n!}\right| \le \frac{M_C}{r^n} \le \frac{M}{r^n}$$

donde M, la cota superior sobre $|f(x)|$ para todo valor de z, la cual, por hipótesis, existe y es independiente de r. Como r se puede tomar arbitrariamente grande, se infiere, por tanto, que el coeficiente de z^n es cero para $n = 1, 2, 3,\ldots$, en otras palabras, para todo valor de z,

$$f(z) = f(0)$$

lo cual prueba el teorema.

www.FreeLibros.me

Una función que es analítica para todo valor de z se conoce como función entera o función integral y, así, el teorema de Liouville afirma que cualquier función entera que es acotada para todos los valores de z, es constante.

1.4 Código binario

La forma de expresión numérica de uso cotidiano se denomina sistema decimal, debido a que su base es el diez, así es como se expresan todas las operaciones matemáticas. Sin embargo, la forma de almacenar de una computadora es en sistema binario (código binario), es decir, sólo utiliza el 0 y el 1 para representar cualquier número [Maron *et al.*, 1995], [Mathews *et al.*, 1992]. Según sea el diseño del hardware de una computadora será la precisión con la que se represente un número. Por ejemplo, para una computadora que utiliza 8 bytes, es decir, 64 bits para representar un entero, el primer bit lo utiliza para el signo y los restantes 63 para el número, así el máximo valor que se puede representar será

$$\sum_{k=0}^{62} 2^k \tag{1.5}$$

El formato para la representación de un número real en una computadora depende del diseño de hardware y software. El formato común es el de punto flotante, donde se le asignan ciertos bits para guardar el exponente y el resto para la mantisa. Por último, la causa fundamental de errores en una computadora se atribuye al error de representar un número real mediante un número limitado de bits, comúnmente se denominan *errores de redondeo* al guardar un número en la memoria.

La épsilon, ε, de una computadora es lo que determina la precisión de la representación de un número y, está definida como el tamaño del intervalo entre 1 y el siguiente número mayor que 1 distinguible de 1. Esto significa que ningún número entre 1 y $1 + \varepsilon$ se puede representar en la computadora. Por ejemplo, cualquier número $1 + \alpha$ se redondea a 1 si $0 < \alpha < \varepsilon/2$, o se redondea a $1 + \varepsilon$ si $\varepsilon/2 \le \alpha$. Así, se puede considerar que $\varepsilon/2$ es el máximo error posible de redondeo para 1. En otras palabras, cuando se halla 1.0 en la memoria de la computadora, el valor original pudo estar entre $1-\varepsilon/2 < x < 1+\varepsilon/2$.

Finalmente se puede concluir que el error de redondeo implicado al guardar cualquier número real R en la memoria de una computadora, es aproximadamente igual a $\varepsilon R/2$, si el número se redondea por exceso y εR si se redondea por defecto.

Las cuatro operaciones aritméticas básicas suma, resta, multiplicación y división en notación binaria presentan dos tipos de situaciones en los que aparecen muchos errores de redondeo:

- Cuando se suma o se resta un número muy pequeño con uno muy grande.
- Cuando se divide entre un número pequeño o se multiplica por un número muy grande.

Para ejemplificar cómo surgen los errores de redondeo se considera el cálculo de 1 y 0.00001 con una mantisa de 24 bits; por tanto, en base 2 se tiene que:

$$(1)_{10} = (0.1000\ \ 0000\ \ 0000\ \ 0000\ \ 0000\ \ 0000)_2 \times 2^1$$

$$(0.00001)_{10} = (0.1010\ \ 0111\ \ 1100\ \ 0101\ \ 1010\ \ 1100)_2 \times 2^{-16}$$

La suma de estos dos números es:

$$(1)_{10} + (0.00001)_{10} = (0.1000\ \ 0000\ \ 0000\ \ 0000\ \ 0101\ \ 0011\ \ \mathbf{1110\ 0010\ 1101\ 0110\ 0})_2 \times 2^1$$

Como se considera una mantisa de sólo 24 bits, se redondea a partir de los números en negrita. Por tanto, el resultado de este cálculo se guarda en la memoria como

$$(1)_{10} + (0.00001)_{10} \cong (0.1000\ \ 0000\ \ 0000\ \ 0000\ \ 0101\ \ 0011)_2 \times 2^1$$

que es equivalente a $(1.0000\ \ 1001\ \ 36)_{10}$.

www.FreeLibros.me

Así, siempre que se sumen 1 y 0.00001, el resultado agrega 0.0000000136 como error. En las operaciones fundamentales, algunas cantidades se redondean por exceso y otros por defecto. La pérdida y ganancia se conoce como *error de redondeo*.

1.5 Números en representación de punto flotante y su aritmética

La forma estándar de representar un número real en forma decimal es con una parte entera, un punto decimal y una parte fraccionaria, por ejemplo 45.67432 o 0.00012534. Otra forma estándar, llamada *notación científica normalizada*, en la cual la parte entera es cero y se obtiene al multiplicar por una potencia adecuada de 10. Se tiene que 45.67432 se puede representar en notación científica normalizada como 0.4567432×10^2 y 0.0012534 como 0.12534×10^{-2}.

En notación científica, un número se representa por una fracción multiplicada por una potencia de 10 y el primer número a la derecha del punto decimal es diferente de cero. En computación a la notación científica normalizada se le llama representación de punto flotante.

Un sistema computacional representa a los números en punto flotante con las limitaciones que impone una longitud finita de palabra. Una computadora puede tener una longitud de palabra de 64 bits (dígitos binarios) la cual puede representar números binarios en la forma $x = \pm\, q \times 2^m$ donde el signo de x ocupa 1 bit; el signo de m, 1 bit; el entero $|m|$, 10 bits y el número q, 52 bits. Al número q se le denomina mantisa y al número m se le llama exponente. Si se supone que m se puede representar con a lo más *10* bits entonces el máximo valor de m que se puede representar es $2^{10} - 1 = 1\,023$.

Los números reales representables en una computadora se llaman *números de máquina*. Para poder representar un número $x = 0.d_1 d_2 d_{3\ldots} \times 10^m$ como número de máquina, denotado por $fl(x)$, se obtiene cortando la mantisa de x en k cifras. Existen dos formas de efectuar ese corte. La primera es simplemente eliminando los dígitos a partir de d_{k+1}. Esta forma recibe el nombre de *truncamiento*. La segunda forma es determinar a d_k a partir de d_{k+1}, si $d_{k+1} < 5$, d_k queda sin cambio. En caso contrario aumentamos en uno d_k, esto se conoce como *redondeo*.

En el análisis numérico es importante tener en cuenta que las soluciones calculadas son simples aproximaciones a las soluciones exactas. La precisión de una solución numérica disminuye en el transcurso de los cómputos. A fin de analizar un poco este problema se define el error de aproximación. Para esto, se supone que $\bar{p}$ es una aproximación de p, se define el *error de aproximación* como $E = p - \bar{p}$ y el *error relativo* así, $E_r = \dfrac{p-\bar{p}}{p}$, si $p \neq 0$.

EJEMPLO 1.3

Si $p = 3.14159$ y $\bar{p} = 3.14$ se tiene que $E = 3.14159 - 3.14 = 0.00159$ y $E_r = \dfrac{3.14159 - 3.14}{3.14159} = \dfrac{0.00159}{3.14159} = 0.000506$.

Si $p = 1\,000$ y $\bar{p} = 998$ se tiene que $E = 2$ y $E_r = \dfrac{2}{1000} = 0.002$. Finalmente, $p = 0.0000001$ y $\bar{p} = 0.000000097$ se tiene que $E = 0.000000003$ y $E_r = \dfrac{0.000000003}{0.0000001} = 0.03$.

De estos ejemplos se observa que el error relativo es por lo general un mejor indicador de la precisión de la aproximación.

Para investigar la propagación del error en cálculos sucesivos se consideran dos números p y q junto con sus aproximaciones p y q con errores ε_p y ε_q, respectivamente. Entonces

$$p + q = (\bar{p} + \varepsilon_p) + (\bar{q} + \varepsilon_q) = (\bar{p} + \bar{q}) + (\varepsilon_p + \varepsilon_q)$$

www.FreeLibros.me

por lo que el error de la suma es la suma de los errores y el error relativo está dado por

$$\frac{(p+q)-(\bar{p}+\bar{q})}{(p+q)}=\frac{\varepsilon_p+\varepsilon_q}{p+q}$$

Para la multiplicación se tiene que

$$pq=(\bar{p}+\varepsilon_p)(\bar{q}+\varepsilon_q)=\bar{p}\bar{q}+\bar{p}\varepsilon_q+\bar{q}\varepsilon_p+(\varepsilon_p+\varepsilon_q)$$

Por lo que si $|p|,|q|>1$ los términos $\bar{p}\varepsilon_q$ y $\bar{q}\varepsilon_p$ muestran un aumento en los errores originales. Para tener una mejor visión de la multiplicación, se calcula el error relativo, bajo las suposiciones de $p,q\neq 0,\frac{p}{\bar{p}}\approx 1,\frac{q}{\bar{q}}\approx 1,\frac{\varepsilon_p}{p}\frac{\varepsilon_q}{q}=E_pE_q\approx 0$. De aquí se obtiene

$$\frac{pq-\bar{p}\bar{q}}{pq}=(\bar{p}+\varepsilon_p)(\bar{q}+\varepsilon_q)=\frac{\varepsilon_q}{q}+\frac{\varepsilon_p}{p}+E_pE_q\approx\frac{\varepsilon_q}{q}+\frac{\varepsilon_p}{p}=E_p+E_q$$

Esto es, el error relativo en el producto es aproximadamente la suma de las razones entre los errores y las aproximaciones.

A menudo un error inicial se puede propagar en una sucesión de cálculos. Un algoritmo se dice que es numéricamente estable si los errores iniciales producen pequeños cambios en el resultado final. De otro modo se dice que el algoritmo es inestable.

Las operaciones básicas de la aritmética tienen sus operaciones equivalentes en las operaciones de punto flotante. Hay que observar que estas operaciones tienen un error que se debe tomar en cuenta. Las operaciones con punto flotante correspondientes a las operaciones aritméticas básicas aquí se representan mediante los mismos símbolos, pero encerrados en un círculo o con el símbolo fl antepuesto a la operación.

Para representar un número en la computadora es necesario tener su representación en punto flotante. Para sumar dos números en número flotante, primero hay que igualar los exponentes, sumar las mantisas y luego normalizar el resultado. Considere el siguiente ejemplo.

EJEMPLO 1.4

Efectuar los cálculos con sólo tres cifras decimales. Si $x=0.134=0.134\times 10^0$ y $y=0.00345=0.345\times 10^{-2}$ entonces

$$x\oplus y=0.134\oplus 0.00345=0.134\times 10^0+0.00345\times 10^0=0.137\times 10^0\neq 0.13745=x+y.$$

Si se considera la diferencia $x-y$ donde $x=0.467546$ y $y=0.462301$ tienen 6 dígitos de precisión. La diferencia se efectúa con sólo cuatro dígitos. Así que

$$fl(x-y)=fl(fl(0.467546)-fl(0.462301))=0.4675\times 10^0-0.4623\times 10^0=0.0052$$

Se puede observar que el resultado es exacto con los cuatro dígitos, pero no así con los números originales, por tanto el resultado con aritmética flotante sólo cuenta con dos dígitos de precisión. Este fenómeno de pérdida de precisión se conoce como cancelación catastrófica y ocurre cuando se restan dos números muy cercanos entre sí.

Sea ε la épsilon de la computadora, se puede demostrar que $fl(x+y)=(x+y)(1+\delta)$, $|\delta|<\varepsilon$ y que $fl((x+y)+z)\neq fl((x+(y+z))$. De este último resultado se concluye que la suma (y la multiplicación) no son asociativas, y que siempre que se pueda se deben sumar números del mismo orden de magnitud. Esto se puede observar al calcular la suma $\sum_{n=1}^{5\,000}\frac{1}{n}$. Si la suma se hace del menor al mayor número, es decir, iniciando con $n=5\,000$ se obtendrá un mejor resultado que iniciando en la forma convencional como se muestra en el siguiente ejemplo.

www.FreeLibros.me

EJEMPLO 1.5

Calcular la suma $\sum_{n=1}^{5\,000}\frac{1}{n}$ utilizando sólo 5 cifras significativas. Si se calcula la suma en la forma $1+\frac{1}{2}+\frac{1}{3}+\cdots$, se obtiene: $\sum_{n=1}^{5\,000}\frac{1}{n}=8.8668$ al usar truncamiento y $\sum_{n=1}^{5\,000}\frac{1}{n}=9.0840$ al usar redondeo. Al calcular la suma en la forma $\frac{1}{5\,000}+\frac{1}{4\,999}+\frac{1}{4\,998}+\cdots$, se obtiene: $\sum_{n=1}^{5\,000}\frac{1}{n}=8.8696$ por truncamiento y $\sum_{n=1}^{5\,000}\frac{1}{n}=9.0957$ por redondeo. El resultado exacto con cinco cifras significativas es $\sum_{n=1}^{5\,000}\frac{1}{n}=9.0945$.

En cualquiera de los casos, ya sea por truncamiento o por redondeo, se observa que al sumar números de igual magnitud se reduce el error de redondeo. Esta diferencia se acentúa al incrementar el número de términos usados en la suma.

A continuación se lista una serie de recomendaciones para reducir el error de redondeo

1. Cuando se van a sumar o restar números, considerar siempre los números más pequeños primero.
2. Evitar en lo posible la resta de dos números aproximadamente iguales. De ser posible reescribirla.
3. Evitar la división entre un número relativamente pequeño.
4. Minimizar el número de operaciones aritméticas en cualquier algoritmo.
5. Programar en precisión simple y doble precisión y comparar los resultados obtenidos.

Ejemplos de esto son las evaluaciones de las funciones $x - \operatorname{sen}(x)$ y $\frac{x^3}{6}$ para x cercano a cero (la segunda expresión es la serie de Taylor truncada). La segunda expresión tiene mejores propiedades numéricas que la primera y se obtienen mejores resultados numéricos.

EJEMPLO 1.6

Al evaluar la función $f(x)=x-\operatorname{sen}(x)$ se tiene que, cerca del 0, los dos términos que la componen son muy similares y pequeños. Para evitar restarlos se obtiene su serie de Taylor $f(x)=x-\operatorname{sen}(x)=\frac{x^3}{3!}-\frac{x^5}{5!}+\frac{x^7}{7!}+\cdots$. Considerando sólo el primer término de la serie $\frac{x^3}{3!}$ y utilizando sólo 5 cifras significativas para evaluar $f(0.0123)$, se tiene que $0.0123 - \operatorname{sen}(0.01213) = 0$, mientras que $\frac{0.0123^3}{3!}=0.31014\times10^{-6}$, el cual es el valor exacto con cinco cifras significativas.

También en la evaluación de polinomios se obtiene una mejor aproximación al usar el método de Hörner, o fórmula anidada, para evaluarlos como se puede ver en el polinomio

$$p(x)=x^3-3x^2+5x-3=(((x-3)x+5)x-3)$$

EJEMPLO 1.7

Al evaluar el polinomio $p(x)=x^3-3x^2+5x-3$ en $x = 1.085$ con aritmética de 6 cifras significativas se tiene que: $p(1.085) = -0.00010$ si se usa truncamiento y $p(1.085) = -0.00010$ si se usa redondeo. Al escribir el polinomio en forma anidada (método de Hörner) se tiene $q(x)=(((x-3)x+5)x-3)$, de donde $q(1.085) = -0.00011$ con truncamiento y $q(1.085) = -0.00011$ usando redondeo. El valor exacto es $p(1.085) = -0.000108375$, esto permite ver que la forma anidada reduce el error. Este problema se agudiza conforme se acerca a las raíces del polinomio, ya que se presenta el fenómeno de cancelación catastrófica. Por lo anterior se recomienda que siempre se escriban los polinomios en forma anidada, ya que mejoran la precisión al reducir el número de operaciones.

www.FreeLibros.me

Para obtener cotas para el error cometido es aconsejable trabajar *a posteriori*, es decir, calcular las cotas junto con la aplicación del algoritmo conforme se vaya calculando el resultado. Cabe hacer notar que esto no siempre es sencillo, pero es la forma más simple de trabajar.

Problemas propuestos

1.6.1 Calcule la serie de Taylor en $x_0 = 0$ de $f(x) = x^4 - 3x^2 + 2$.

1.6.2 Calcule el cuarto polinomio de Taylor en $x_0 = 0$ de $f(x) = \sqrt{x+1}$ y utilícelo para aproximar a $\sqrt{2}$. Calcule el error real cometido.

1.6.3 El n-ésimo polinomio de Taylor de $g(x) = e^x$ está dado por $p_n(x) = 1 + \frac{1}{1!}x + \frac{1}{2!}x^2 + \frac{1}{3!}x^3 + \cdots + \frac{1}{n!}x^n$. El $2n$-ésimo polinomio de Taylor de $f(x) = e^{-x^2}$ se puede obtener como $p_n(-x^2)$. A partir de esto aproxime la integral $\int_0^1 e^{-x^2}dx$ usando el octavo polinomio de $f(x) = e^{-x^2}$. Determine el error real cometido.

1.6.4 Calcule el onceavo polinomio de Taylor en $x_0 = 0$ de $f(x) = \arctan(x)$ y utilice el hecho de que $\arctan(1) = \frac{\pi}{4}$ para aproximar a π. Calcule el error real cometido.

1.6.5 Calcule el error y el error relativo de las aproximaciones de p mediante $\bar{p}$. 1) $p = \pi$, $\bar{p} = 3.1416$, 2) $p = \pi$, $\bar{p} = 22/7$, 3) $p = e$, $\bar{p} = 2.8182$ y 4) $p = e$, $\bar{p} = 65/24$.

1.6.6 Calcule $\frac{122}{135} - \frac{11}{32} + \frac{20}{19}$ mediante aritmética exacta, utilice truncamiento a tres cifras y redondeo hasta tres cifras. Determine los errores y los errores relativos.

1.6.7 Las expresiones 215 – 0.345 – 214 y 215 – 214 – 0.345 son idénticas. Calcule mediante aritmética exacta, emplee truncamiento y redondeo hasta tres cifras. Determine los errores y los errores relativos. Este problema muestra que el orden en que se operan los términos sí afecta el resultado.

1.6.8 El sistema de ecuaciones lineales

$$\begin{aligned} 19.84x + 24.35y &= 68.54 \\ 6.98x + 8.32y &= 23.62, \end{aligned}$$

tiene solución $x = 1$, $y = 2$. Una forma de resolver este sistema es utilizando la regla de Cramer para sistemas de dos ecuaciones. La regla de Cramer para este sistema establece que

$$x = \frac{(23.62)(24.35) - (68.54)(8.32)}{(6.98)(24.35) - (19.84)(8.32)}$$

y

$$y = \frac{(6.98)(68.54) - (19.84)(23.62)}{(6.98)(24.35) - (19.84)(8.32)} .$$

Calcule mediante aritmética exacta, truncamiento y redondeo hasta tres cifras estas expresiones. Determine los errores y los errores relativos.

1.6.9 Dado el sistema

$$\begin{aligned} -37.64x + 42.11y &= -28.7 \\ 3.48x + 5.84y &= 22.12. \end{aligned}$$

Use la regla de Cramer para resolver el sistema usando aritmética exacta y redondeo con tres cifras decimales.

www.FreeLibros.me

1.6.10 La función $f(x)=\sqrt{x^2+1}-1$ presenta cancelación catastrófica para $x \approx 0$. Evalúela en $x=0.01$ con aritmética exacta, truncamiento y redondeo con tres cifras. Calcule sus errores y errores relativos. Multiplicando $f(x)$ por $\frac{\sqrt{x^2+1}+1}{\sqrt{x^2+1}+1}$ se tiene que $f(x)=\frac{x^2}{\sqrt{x^2+1}+1}$. Utilice esta forma para evaluar a $f(x)$ en el punto dado. Comente sus resultados.

1.6.11 Evalúe la función $f(x)=\frac{e^x-(1+x)}{x^2}$ en $x=0.01$ con aritmética exacta, truncamiento y redondeo con tres cifras. Calcule sus errores y errores relativos. Usando la serie de Taylor de e^x en $x_0=0$ se obtiene $f(x)=\frac{1}{2!}+\frac{1}{3!}x+\frac{1}{4!}x^2+\frac{1}{5!}x^3+\cdots$. Utilice el polinomio de Taylor de primer orden para aproximar a $f(x)$ en $x=0.01$ con aritmética exacta, truncamiento y redondeo con tres cifras decimales. Determine el error de aproximación.

1.6.12 Use la fórmula anidada de $p(x)=x^5-7x^4-x^3+4x^2-2x+1$ para evaluar $p(-1.5)$.

1.6.13 Evalúe el polinomio $p(x)=x^3+4.12x^2-3.16x+1.34$ en $x=-4.831$ con aritmética exacta, truncamiento y redondeo con cuatro cifras. Calcule sus errores y errores relativos. Evalúe usando anidamiento y repita los cálculos.

1.6.14 Al resolver la ecuación cuadrática $ax^2+bx+c=0$ se pueden presentar serios problemas de cancelación catastrófica si una de las raíces es muy pequeña en comparación con la otra. A fin de evitar este problema en lugar de usar la fórmula

$$r_1, r_2 = \frac{-b\pm\sqrt{b^2-4ac}}{2a}$$

se usan las fórmulas

$$r_1 = signo\left(\frac{-b}{2a}\right)\left[\left|\frac{-b}{2a}\right|+\sqrt{\left(\frac{b}{2a}\right)^2-\frac{c}{a}}\right]; r_2=\frac{c/a}{r_1}.$$

Utilice los dos métodos para resolver la ecuación $x^2-10^5x+1=0$ usando truncamiento con 7 cifras

1.6.15 Resuelva la ecuación $x^2+1000.001x+1$ usando los métodos propuestos en el problema anterior.

1.6.16 Evalúe la suma $\sum_{i=1}^{60}\frac{(-12)^i}{i!}$ truncamiento con 10 cifras en el orden usual. Compare el resultado obtenido sumando en sentido inverso. El valor real de la suma con 10 cifras es $\sum_{i=1}^{60}\frac{(-12)^i}{i!}=0.6144212353\times10^{-5}$.

www.FreeLibros.me

www.FreeLibros.me

Capítulo 2

Solución de ecuaciones no lineales

2.1 Introducción

Existen ecuaciones de tipo *no lineal* (no polinomiales) que son muy conocidas debido a que existen algunos métodos analíticos que conducen a una fórmula para su solución; como por ejemplo, la solución de ciertas ecuaciones trigonométricas simples. Sin embargo, muchas ecuaciones no lineales no se pueden resolver directamente por métodos analíticos, por lo que se deben usar métodos basados en aproximaciones numéricas. En este capítulo se consideran algunos métodos numéricos para determinar las soluciones reales de ecuaciones de la forma

$$f(x)=0 \tag{2.1}$$

donde f es una función real. Para hacer esto se consideran métodos iterativos con el fin de resolver la ecuación (2.1); esto es, dado un valor inicial x_0. se construye una sucesión de números reales $\{x_n\}_{n=0}^{\infty}=\{x_0, x_1, x_2,\ldots\}$. Si la ecuación (2.1) y el método definido son adecuados, se espera que $|x_n - x^*| \to 0$ cuando $n \to \infty$. En este caso se dice que el método *converge* a la solución x^*; de no ser así, se dice que el método *diverge*. Este capítulo se inicia con el método de bisección, ya que este es el más fácil de entender y aplicar.

2.2 Método de bisección

La solución del problema se plantea en términos de que se necesita encontrar el valor de x tal que $f(x)=0$, *un cruce por cero*. Para iniciar el método, se determinan dos puntos x_0 y $\tilde{x}_0$ en los cuales la función toma valores con signo opuesto. Si se supone que la función es continua, con el teorema del valor intermedio se garantiza que debe existir al menos un cruce por cero de f entre x_0 y $\tilde{x}_0$. La función se evalúa entonces en el punto $x_1=\frac{1}{2}(x_0+\tilde{x}_0)$. El punto $\tilde{x}_1$ se elige como análogo del par $(x_0, \tilde{x}_0)$ en el cual el valor de la función tenga signo opuesto a $f(x_1)$. Se obtiene así un intervalo $[x_1, \tilde{x}_1]$ que continúa conteniendo un cruce por cero y que tiene la mitad del tamaño del intervalo original. El proceso se repite hasta que los límites superior e inferior del cruce por cero estén suficientemente cercanos. La figura 2.1 muestra un paso típico de este método. El método se

www.FreeLibros.me

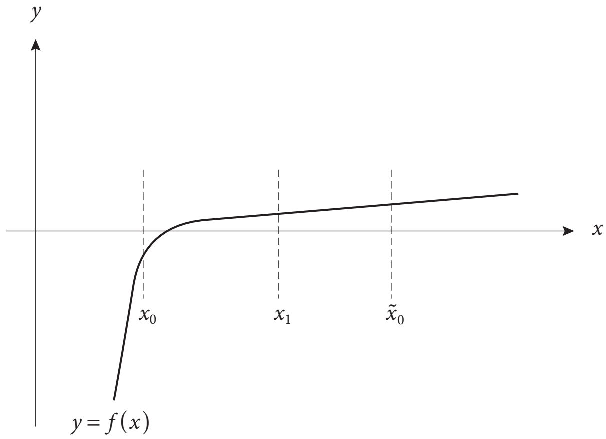


Figura 2.1 Método de bisección.

detiene cuando $|x_n - \tilde{x}_n| < tol$ o $|f(x_{n+1})| < tol$, donde *tol* es la tolerancia especificada para el método de bisección [Nakamura, 1992], [Maron, 1995], [Burden *et al.*, 2002], [Nieves *et al.*, 2002].

El método de bisección tiene la ventaja de ser sencillo; sin embargo, la velocidad de convergencia es lenta y, cuando las iteraciones se aproximan al cruce por cero, es mejor usar otro método que converja más rápido. En la sección 2.10.1 se presenta un programa desarrollado en Matlab para este método. La convergencia del método de bisección sólo se asegura a partir de los siguientes resultados.

Teorema 2.1 Suponiendo que *f* es continua en el intervalo $[x_0, \tilde{x}_0]$ y $f(x_0)f(\tilde{x}_0) < 0$, el método de bisección genera una sucesión $\{x_n\}_{n=1}^{\infty}$ que aproxima x^* a un cruce por cero de *f*, tal que

$$|x_n - x^*| \leq \frac{x_0 - \tilde{x}_0}{2^n}, \quad n \geq 1$$

Este último teorema implica que

$$x_n = x^* + O\left(\frac{1}{2^n}\right)$$

donde $O(1/2^n)$ es la velocidad de convergencia del método. Se tiene además que el método de bisección satisface

$$\lim_{n\to\infty} \frac{|x_{n+1} - x^*|}{|x_n - x^*|} = \frac{1}{2} \tag{2.2}$$

por lo que el método converge linealmente.

EJEMPLO 2.1

Considerando la ecuación $f(x) = 1 + 2x - 3x^2e^{-x} + 2x^3 \operatorname{sen}(x)e^{-\frac{x}{5}}$, calcular los cruces por cero dentro del intervalo $[4, 20]$; usar el método de bisección con un error de 10^{-5}.

Solución. En primer lugar se gráfica la función para tener una idea del comportamiento dentro del intervalo especificado y de cuántas veces se cruza por cero. Analizando esta figura, a simple vista se puede determinar que cruza cinco veces por cero y es continua en todo el intervalo. Por tanto, el método de bisección es adecuado para determinar estos cruces por cero.

www.FreeLibros.me

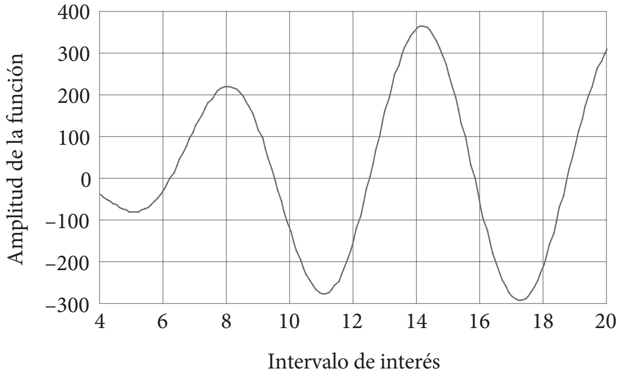


Figura 2.2 Gráfica de la función $f(x)$.

La tabla 2.1 provee de todos los pasos intermedios hasta encontrar la convergencia. Los intervalos donde se encuentran los cruces se pueden visualizar en la figura 2.2; éstos se pueden encontrar numéricamente de manera muy sencilla, evaluando la función y tomando dos puntos consecutivos en donde la función cambia de signo.

En la sección 2.10.1 se tiene un programa codificado en Matlab que encuentra los intervalos que contienen cada cruce por cero y después aplica el método de bisección para determinar en forma específica cada uno de ellos.

Se debe precisar que, una vez que se tiene un intervalo y una función continua dentro de ese intervalo, este método siempre converge. Sin embargo, no se tiene la garantía de que en cada paso consecutivo el error decrezca; por esta razón es un método seguro, pero de convergencia lenta.

Tabla 2.1a Tabla de valores que se obtienen al usar el método de bisección al calcular el primer cruce por cero.

n	x_n	$f(x_n)$	$\tilde{x}_n$	$f(\tilde{x}_n)$	x_{n+1}	$f(x_{n+1})$
0	6.000000	−23.624104	7.000000	126.005431	6.500000	46.010338
1	6.000000	−23.624104	6.500000	46.010338	6.250000	8.632171
2	6.000000	−23.624104	6.250000	8.632171	6.125000	−8.262447
3	6.125000	−8.262447	6.250000	8.632171	6.187500	0.007349
4	6.125000	−8.262447	6.187500	0.007349	6.156250	−4.173800
5	6.156250	−4.173800	6.187500	0.007349	6.171875	−2.094560
6	6.171875	−2.094560	6.187500	0.007349	6.179687	−1.046410
7	6.179687	−1.046410	6.187500	0.007349	6.183593	−0.520228
8	6.183593	−0.520228	6.187500	0.007349	6.185546	−0.256613
9	6.185546	−0.256613	6.187500	0.007349	6.186523	−0.124675
10	6.186523	−0.124675	6.187500	0.007349	6.187011	−0.058673
11	6.187011	−0.058673	6.187500	0.007349	6.187255	−0.025664
12	6.187255	−0.025664	6.187500	0.007349	6.187377	−0.009158
13	6.187377	−0.009158	6.187500	0.007349	6.187438	−0.000904
14	6.187438	−0.000904	6.187500	0.007349	6.187469	0.003222
15	6.187438	−0.000904	6.187469	0.003222	6.187454	0.001158
16	6.187438	−0.000904	6.187454	0.001158	6.187446	0.000127
17	6.187438	−0.000904	6.187446	0.000127	6.187442	−0.000388
18	6.187442	−0.000388	6.187446	0.000127	6.187444	−0.000130
19	6.187444	−0.000130	6.187446	0.000127	6.187445	−1.89869e−6

www.FreeLibros.me

Tabla 2.1b Tabla de valores que se obtienen al usar el método de bisección al calcular el segundo cruce por cero.

n	x_n	$f(x_n)$	$\tilde{x}_n$	$f(\tilde{x}_n)$	x_{n+1}	$f(x_{n+1})$
0	9.000000	118.292948	10.000000	–126.264122	9.500000	0.705516
1	9.500000	0.705516	10.000000	–126.264122	9.750000	–63.785375
2	9.500000	0.705516	9.750000	–63.785375	9.625000	–31.507783
3	9.500000	0.705516	9.625000	–31.507783	9.562500	–15.358134
4	9.500000	0.705516	9.562500	–15.358134	9.531250	–7.311246
5	9.500000	0.705516	9.531250	–7.311246	9.515625	–3.298564
6	9.500000	0.705516	9.515625	–3.298564	9.507812	–1.295382
7	9.500000	0.705516	9.507812	–1.295382	9.503906	–0.294639
8	9.500000	0.705516	9.503906	–0.294639	9.501953	0.205512
9	9.501953	0.205512	9.503906	–0.294639	9.502929	–0.044544
10	9.501953	0.205512	9.502929	–0.044544	9.502441	0.080488
11	9.502441	0.080488	9.502929	–0.044544	9.502685	0.017973
12	9.502685	0.017973	9.502929	–0.044544	9.502807	–0.013285
13	9.502685	0.017973	9.502807	–0.013285	9.502746	0.002343
14	9.502746	0.002343	9.502807	–0.013285	9.502777	–0.005470
15	9.502746	0.002343	9.502777	–0.005470	9.502761	–0.001563
16	9.502746	0.002343	9.502761	–0.001563	9.502754	0.000390
17	9.502754	0.000390	9.502761	–0.001563	9.502758	–0.000586
18	9.502754	0.000390	9.502758	–0.000586	9.502756	–9.82125e–5

Tabla 2.1c Tabla de valores que se obtienen al usar el método de bisección al calcular el tercer cruce por cero.

n	x_n	$f(x_n)$	$\tilde{x}_n$	$f(\tilde{x}_n)$	x_{n+1}	$f(x_{n+1})$
0	12.000000	–143.229664	13.000000	164.123770	12.500000	4.732499
1	12.000000	–143.229664	12.500000	4.732499	12.250000	–73.208212
2	12.250000	–73.208212	12.500000	4.732499	12.375000	–34.926682
3	12.375000	–34.926682	12.500000	4.732499	12.437500	–15.229780
4	12.437500	–15.229780	12.500000	4.732499	12.468750	–5.276783
5	12.468750	–5.276783	12.500000	4.732499	12.484375	–0.278543
6	12.484375	–0.278543	12.500000	4.732499	12.492187	2.225456
7	12.484375	–0.278543	12.492187	2.225456	12.488281	0.973066
8	12.484375	–0.278543	12.488281	0.973066	12.486328	0.347162
9	12.484375	–0.278543	12.486328	0.347162	12.485351	0.034284
10	12.484375	–0.278543	12.485351	0.034284	12.484863	–0.122135
11	12.484863	–0.122135	12.485351	0.034284	12.485107	–0.043927
12	12.485107	–0.043927	12.485351	0.034284	12.485229	–0.004821
13	12.485229	–0.004821	12.485351	0.034284	12.485290	0.014731
14	12.485229	–0.004821	12.485290	0.014731	12.485260	0.004954
15	12.485229	–0.004821	12.485260	0.004954	12.485244	6.65014e–5

www.FreeLibros.me

Tabla 2.1d Tabla de valores que se obtienen al usar el método de bisección al calcular el cuarto cruce por cero.

n	x_n	$f(x_n)$	$\tilde{x}_n$	$f(\tilde{x}_n)$	x_{n+1}	$f(x_{n+1})$
0	15.000000	249.537288	16.000000	−63.137906	15.500000	101.272844
1	15.500000	101.272844	16.000000	−63.137906	15.750000	18.428240
2	15.750000	18.428240	16.000000	−63.137906	15.875000	−22.850348
3	15.750000	18.428240	15.875000	−22.850348	15.812500	−2.293390
4	15.750000	18.428240	15.812500	−2.293390	15.781250	8.052087
5	15.781250	8.052087	15.812500	−2.293390	15.796875	2.874856
6	15.796875	2.874856	15.812500	−2.293390	15.804687	0.289527
7	15.804687	0.289527	15.812500	−2.293390	15.808593	−1.002242
8	15.804687	0.289527	15.808593	−1.002242	15.806640	−0.356434
9	15.804687	0.289527	15.806640	−0.356434	15.805664	−0.033472
10	15.804687	0.289527	15.805664	−0.033472	15.805175	0.128023
11	15.805175	0.128023	15.805664	−0.033472	15.805419	0.047274
12	15.805419	0.047274	15.805664	−0.033472	15.805541	0.006900
13	15.805541	0.006900	15.805664	−0.033472	15.805603	−0.013285
14	15.805541	0.006900	15.805603	−0.013285	15.805572	−0.003192
15	15.805541	0.006900	15.805572	−0.003192	15.805557	0.001854
16	15.805557	0.001854	15.805572	−0.003192	15.805564	−0.000669
17	15.805557	0.001854	15.805564	−0.000669	15.805561	0.000592
18	15.805561	0.000592	15.805564	−0.000669	15.805562	−3.83656e−5

Tabla 2.1e Tabla de valores que se obtienen al usar el método de bisección al calcular el quinto cruce por cero.

n	x_n	$f(x_n)$	$\tilde{x}_n$	$f(\tilde{x}_n)$	x_{n+1}	$f(x_{n+1})$
0	18.000000	−202.342578	19.000000	84.994649	18.500000	−69.223908
1	18.500000	−69.223908	19.000000	84.994649	18.750000	7.683801
2	18.500000	−69.223908	18.750000	7.683801	18.625000	−31.130974
3	18.625000	−31.130974	18.750000	7.683801	18.687500	−11.774935
4	18.687500	−11.774935	18.750000	7.683801	18.718750	−2.053533
5	18.718750	−2.053533	18.750000	7.683801	18.734375	2.813750
6	18.718750	−2.053533	18.734375	2.813750	18.726562	0.379686
7	18.718750	−2.053533	18.726562	0.379686	18.722656	−0.837038
8	18.722656	−0.837038	18.726562	0.379686	18.724609	−0.228703
9	18.724609	−0.228703	18.726562	0.379686	18.725585	0.075484
10	18.724609	−0.228703	18.725585	0.075484	18.725097	−0.076611
11	18.725097	−0.076611	18.725585	0.075484	18.725341	−0.000563
12	18.725341	−0.000563	18.725585	0.075484	18.725463	0.037460
13	18.725341	−0.000563	18.725463	0.037460	18.725402	0.018448
14	18.725341	−0.000563	18.725402	0.018448	18.725372	0.008942
15	18.725341	−0.000563	18.725372	0.008942	18.725357	0.004189
16	18.725341	−0.000563	18.725357	0.004189	18.725349	0.001812
17	18.725341	−0.000563	18.725349	0.001812	18.725345	0.000624
18	18.725341	−0.000563	18.725345	0.000624	18.725343	3.054369e−5

www.FreeLibros.me

2.3 Método de la falsa posición o regla falsa

Una modificación simple del método de bisección produce otro método que siempre es convergente [Nakamura, 1992], [Maron, 1995], [Burden *et al.*, 2002], [Nieves *et al.*, 2002]. Si se pueden elegir dos aproximaciones iniciales x_0 y $\tilde{x}_0$ tales que los dos valores de la función en esos puntos tengan signo opuesto, entonces es posible generar una sucesión de valores que siempre tengan esta propiedad. Para iniciar, se construye la recta que pasa por los puntos $(x_0, f(x_0))$ y $(\tilde{x}_0, f(\tilde{x}_0))$. De acuerdo con la figura 2.3, se tiene que $m_1 = m_2$; por tanto:

$$\frac{f(x_1)-f(x_0)}{x_1-x_0}=\frac{f(\tilde{x}_0)-f(x_0)}{\tilde{x}_0-x_0} \tag{2.3a}$$

El valor del cruce por cero se define cuando se tiene un valor de x_1, dado por la recta definida por la ecuación (2.3a), donde se cumple que $f(x_1)=0$. Así, la ecuación anterior queda de la siguiente forma

$$\frac{0-f(x_0)}{x_1-x_0}=\frac{f(\tilde{x}_0)-f(x_0)}{\tilde{x}_0-x_0}$$

Despejando x_1 se obtiene

$$x_1 = x_0 - \frac{f(x_0)(\tilde{x}_0-x_0)}{f(\tilde{x}_0)-f(x_0)} \tag{2.3b}$$

Utilizando la ecuación (2.3b), el valor de $\tilde{x}_1$ se elige tomando un valor entre x_0 y $\tilde{x}_0$ de tal forma que el valor de la función sea opuesto en signo a $f(x_1)$. Así, valores de x_1 y $\tilde{x}_1$ definen un menor intervalo que contiene el cruce por cero. El proceso continua tomando siempre lados opuestos del cruce por cero. La penalidad que ocasiona esta modificación del método de bisección es el número de operaciones necesarias para calcular los valores de $\{x_n\}_{n=1}^{\infty}$. La longitud de los nuevos intervalos, para el *método de falsa posición*, no decrece en cada nueva iteración como en el método de bisección, es decir, no siempre se garantiza que el nuevo intervalo sea la mitad (o menor) del intervalo anterior. Por esta razón, aunque el método de la falsa posición normalmente tiene una mejor convergencia que el método de bisección, no siempre será el caso. El programa desarrollado en Matlab para el método de falsa posición se proporciona en la sección 2.10.2.

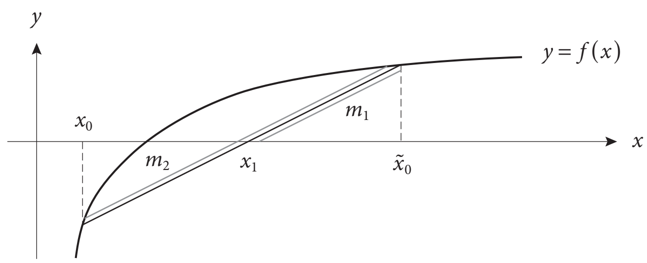


Figura 2.3 Método de la falsa posición.

EJEMPLO 2.2

Considerando la misma ecuación que en el ejemplo 2.1, es decir, $f(x)=1+2x-3x^2e^{-x}+2x^3\operatorname{sen}(x)e^{-\frac{x}{5}}$; calcular los cruces por cero dentro del intervalo $[4, 20]$; usar el método de *regla falsa* o falsa posición con un error máximo de 10^{-5}.

SOLUCIÓN. La tabla 2.2 muestra todos los resultados del cálculo de los cruces por cero.

www.FreeLibros.me

Tabla 2.2a Tabla de valores que se obtienen al usar el método de falsa posición al calcular el primer cruce por cero.

n	x_n	$f(x_n)$	$\tilde{x}_n$	$f(\tilde{x}_n)$	x_{n+1}	$f(x_{n+1})$
0	6.000000	–23.624104	7.000000	126.005431	6.157883	–3.957441
1	6.157883	–3.957441	7.000000	126.005431	6.183526	–0.529249
2	6.183526	–0.529249	7.000000	126.005431	6.186941	–0.068115
3	6.186941	–0.068115	7.000000	126.005431	6.187381	–0.008721
4	6.187381	–0.008721	7.000000	126.005431	6.187437	–0.001115
5	6.187437	–0.001115	7.000000	126.005431	6.187444	–0.000142
6	6.187444	–0.000142	7.000000	126.005431	6.187445	–1.82674e–5

Tabla 2.2b Tabla de valores que se obtienen al usar el método de falsa posición al calcular el segundo cruce por cero.

n	x_n	$f(x_n)$	$\tilde{x}_n$	$f(\tilde{x}_n)$	x_{n+1}	$f(x_{n+1})$
0	9.000000	118.292948	10.000000	–126.264122	9.483702	4.871535
1	9.483702	4.871535	10.000000	–126.264122	9.502882	–0.032504
2	9.483702	4.871535	9.502882	–0.032504	9.502755	4.91275e–5

Tabla 2.2c Tabla de valores que se obtienen al usar el método de falsa posición al calcular el tercer cruce por cero.

n	x_n	$f(x_n)$	$\tilde{x}_n$	$f(\tilde{x}_n)$	x_{n+1}	$f(x_{n+1})$
0	12.000000	–143.229664	13.000000	164.123770	12.466009	–6.151974
1	12.466009	–6.151974	13.000000	164.123770	12.485302	0.018546
2	12.466009	–6.151974	12.485302	0.018546	12.485244	–3.02743e–5

Tabla 2.2d Tabla de valores que se obtienen al usar el método de falsa posición al calcular el cuarto cruce por cero.

n	x_n	$f(x_n)$	$\tilde{x}_n$	$f(\tilde{x}_n)$	x_{n+1}	$f(x_{n+1})$
0	15.000000	249.537288	16.000000	–63.137906	15.798071	2.478624
1	15.798071	2.478624	16.000000	–63.137906	15.805699	–0.045230
2	15.798071	2.478624	15.805699	–0.045230	15.805562	–1.99405e–5

Tabla 2.2e Tabla de valores que se obtienen al usar el método de falsa posición al calcular el quinto cruce por cero.

n	x_n	$f(x_n)$	$\tilde{x}_n$	$f(\tilde{x}_n)$	x_{n+1}	$f(x_{n+1})$
0	18.000000	–202.342578	19.000000	84.994649	18.704198	–6.582714
1	18.704198	–6.582714	19.000000	84.994649	18.725461	0.036746
2	18.704198	–6.582714	18.725461	0.036746	18.725343	–2.05257e–5

www.FreeLibros.me

2.4 Método de la secante

Un problema obvio que surge con el método de la falsa posición es que, dependiendo de la función, el intervalo de búsqueda puede no decrecer. El problema se evita al considerar los puntos en sucesión estricta en el sentido de que el valor más viejo se descarta, y sólo se usan los dos valores más recientes al calcular el nuevo valor. Esta idea conduce al método de la secante [Maron, 1995], [Burden *et al.*, 2002], [Nieves *et al.*, 2002], [Rodríguez, 2003], [Cordero *et al.*, 2006], el cual se puede deducir de manera simple utilizando la figura 2.4, es decir, si se tiene que $m_1 = m_2$, entonces,

$$\frac{f_1 - f_0}{x_1 - x_0} = \frac{f_2 - f_1}{x_2 - x_1}, \text{ donde se tiene por notación que } f_n = f(x_n).$$

Si se tiene que $f(x_2) = 0$, entonces se llega a

$$x_2 - x_1 = \frac{-(x_1 - x_0) f(x_1)}{f(x_1) - f(x_0)}$$

Reagrupando, se llega finalmente a

$$x_{n+1} = x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})} \tag{2.4}$$

El método de la secante necesita dos valores iniciales x_0 y x_1 para comenzar. Los valores $f(x_0)$ y $f(x_1)$ se calculan y dan dos puntos sobre la curva. El nuevo punto de la sucesión es el punto en el cual la recta que une los dos puntos previos corta al eje x. Si se compara la fórmula (2.4) con la (2.3b), se puede notar que poseen la misma estructura; la única diferencia es la forma de tomar los datos en los pasos de iteración. En el *método de la secante* los puntos se usan en una sucesión estricta. Cada vez que se encuentra un nuevo punto, el número más atrasado se descarta. Al operar de esta forma, se tendrán ciertas iteraciones idénticas a las que se obtienen al aplicar el método de regla falsa; sin embargo, en este método, es totalmente posible que la sucesión diverja como se muestra en la figura 2.5, donde el punto x_2 esta claramente más lejano de la raíz que el punto x_1. La rapidez de convergencia de este método, cuando se está suficientemente cerca de la solución, es superior a la de los métodos de bisección y falsa posición. El programa desarrollado en Matlab para este método se proporciona en la sección 2.10.3.

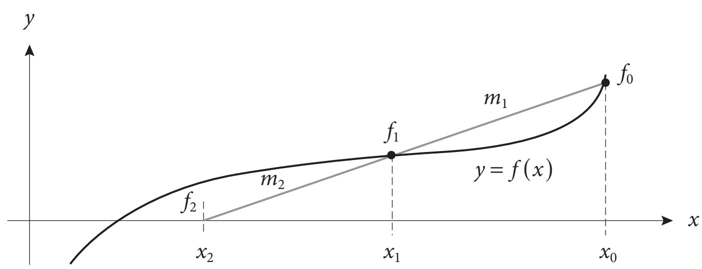


Figura 2.4 Método de la secante.

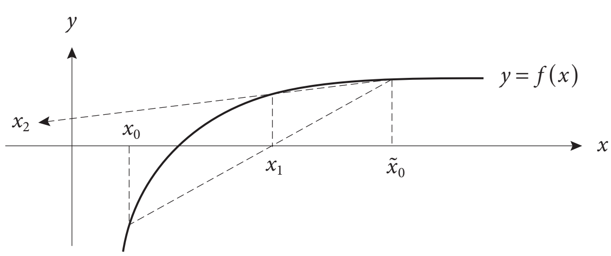


Figura 2.5 Divergencia del método de la secante.

www.FreeLibros.me

La figura 2.5 muestra que el método de la secante puede divergir dependiendo de la naturaleza del problema; sin embargo, antes de aplicarlo, se puede hacer una prueba de convergencia, para lo cual se cuenta con el siguiente teorema que la garantiza.

Teorema 2.2 Suponiendo que f tiene segunda derivada continua, sea x^* tal que $f(x^*)=0$ y $f'(x^*)\neq 0$. Si x_0 es lo suficientemente cercana a x^*, la sucesión $\{x_k\}_{k=0}^{\infty}$ generada por el método de la secante converge a x^* con un orden de convergencia aproximado de $\tau_1 \approx 1.618$.

Demostración Si se define $f[a,b]=\dfrac{f(b)-f(a)}{b-a}$, se tiene que

$$x_{k+1}-x^*=x_k-x^*-f(x_k)\left[\frac{x_k-x_{k-1}}{f(x_k)-f(x_{k-1})}\right]$$

Reagrupando, se obtiene

$$x_{k+1}-x^*=(x_k-x^*)\left\{\frac{f[x_{k-1},x_{x-1}]-f[x_k,x^*]}{f[x_{x-1},x_k]}\right\} \tag{2.5}$$

Definiendo además

$$f[a,b,c]=\frac{f[a,b]-f[b,c]}{a-c} \tag{2.6}$$

se puede escribir la ecuación (2.5) como

$$x_{k+1}-x^*=(x_k-x^*)(x_{k-1}-x^*)\frac{f[x_{k-1},x_k,x^*]}{f[x_{k-1},x_k]}$$

Ahora, del teorema del valor medio se tiene que existen ξ_k entre $[x_k,x_{k-1}]$, η_k entre $[x_{k-1},x_k]$ y x^*, tales que:

$$f[x_{k-1},x_k]=f(\xi_k)$$

y

$$f[x_{k-1},x_k,x^*]=\frac{1}{2}f(\eta_k)$$

Reformulando (2.6), se tiene que

$$x_{k+1}-x^*=\frac{f''(\eta_k)}{2f'(\xi_k)}(x_k-x^*)(x_{k-1}-x^*) \tag{2.7}$$

Se concluye de inmediato que el proceso converge si se inicia lo suficiente cerca de x^*. Para determinar el orden de convergencia, se nota que para k grande la ecuación (2.7) se transforma en

$$x_{k+1}-x^*=M(x_k-x^*)(x_{k-1}-x^*)$$

donde $M=\dfrac{f''(x^*)}{2f'(x^*)}$. Si se define $\varepsilon_k=x_k-x^*$, se tiene que cuando k es lo suficientemente grande, entonces se tiene

$$\varepsilon_{k+1}=M\varepsilon_k\varepsilon_{k-1}$$

Tomando el logaritmo de esta ecuación y haciendo que $y_k=\log M\varepsilon_k$, se obtiene

www.FreeLibros.me

$$y_{k+1} = y_k + y_{k-1}$$

la cual es la ecuación en diferencias de los números de Fibonacci. Una solución a esta ecuación debe satisfacer

$$y_{k+1} - \tau_1 y_k \to 0$$

donde $\tau_1 \approx 1.618$, por lo que

$$\log M\varepsilon_{k+1} - \tau_1 \log M\varepsilon_k \to 0$$

Utilizando una de las propiedades de los logaritmos, se obtiene

$$\log \frac{M\varepsilon_{k+1}}{(M\varepsilon_k)^{\tau_1}} \to 0$$

y por tanto

$$\frac{\varepsilon_{k+1}}{\varepsilon_k} \to M^{(\tau_1 - 1)}$$

EJEMPLO 2.3

Considerando la misma ecuación que en el ejemplo 2.1, es decir, $f(x) = 1 + 2x - 3x^2 e^{-x} + 2x^3 \text{sen}(x) e^{-\frac{x}{5}}$, calcular los cruces por cero dentro del intervalo $[4, 20]$; usar el método de la secante con un error máximo de 10^{-5}.

SOLUCIÓN. Para este ejemplo en particular, la sección 2.10.4 contiene el código Matlab que da la solución completa y, como resultado, se obtiene la tabla 2.3 enunciada a continuación.

Tabla 2.3a Tabla de valores que se obtienen al usar el método de la secante al calcular el primer cruce por cero.

n	x_n	$f(x_n)$	x_{n+1}	$f(x_{n+1})$	x_{n+2}	$f(x_{n+2})$
0	6.000000	–23.624104	7.000000	126.005431	6.157883	–3.957441
1	7.000000	126.005431	6.157883	–3.957441	6.183526	–0.529249
2	6.157883	–3.957441	6.183526	–0.529249	6.187485	0.005409
3	6.183526	–0.529249	6.187485	0.005409	6.187445	–7.15024e–6

Tabla 2.3b Tabla de valores que se obtienen al usar el método de la secante al calcular el segundo cruce por cero.

n	x_n	$f(x_n)$	x_{n+1}	$f(x_{n+1})$	x_{n+2}	$f(x_{n+2})$
0	9.000000	118.292948	10.000000	–126.264122	9.483702	4.871535
1	10.000000	–126.264122	9.483702	4.871535	9.502882	–0.032504
2	9.483702	4.871535	9.502882	–0.032504	9.502755	4.91275e–5

www.FreeLibros.me

Tabla 2.3c Tabla de valores que se obtienen al usar el método de la secante al calcular el tercer cruce por cero.

n	x_n	$f(x_n)$	x_{n+1}	$f(x_{n+1})$	x_{n+2}	$f(x_{n+2})$
0	12.000000	−143.229664	13.000000	164.123770	12.466009	−6.151974
1	13.000000	164.123770	12.466009	−6.151974	12.485302	0.018546
2	12.466009	−6.151974	12.485302	0.018546	12.485244	−3.02743e−5

Tabla 2.3d Tabla de valores que se obtienen al usar el método de la secante al calcular el cuarto cruce por cero.

n	x_n	$f(x_n)$	x_{n+1}	$f(x_{n+1})$	x_{n+2}	$f(x_{n+2})$
0	15.000000	249.537288	16.000000	−63.137906	15.798071	2.478624
1	16.000000	−63.137906	15.798071	2.478624	15.805699	−0.045230
2	15.798071	2.478624	15.805699	−0.045230	15.805562	−1.99405e−5

Tabla 2.3e Tabla de valores que se obtiene al usar el método de la secante al calcular el quinto cruce por cero.

n	x_n	$f(x_n)$	x_{n+1}	$f(x_{n+1})$	x_{n+2}	$f(x_{n+2})$
0	18.000000	−202.342578	19.000000	84.994649	18.704198	−6.582714
1	19.000000	84.994649	18.704198	−6.582714	18.725461	0.036746
2	18.704198	−6.582714	18.725461	0.036746	18.725343	−2.05257e−5

2.5 Método del punto fijo

El *método del punto fijo* es fácil de usar y se aplica a una amplia variedad de problemas. En su forma más simple, la ecuación que se va a iterar se obtiene reagrupando la ecuación que contiene x en el lado izquierdo de la ecuación. Una aproximación a x se inserta en el lado derecho; así se calcula un nuevo valor de x. El nuevo valor de x se usa en el cálculo para dar más valores de x, y el proceso se repite en forma iterativa. Si el punto inicial y la reordenación de la ecuación son adecuados, los valores se aproximan cada vez más a la solución verdadera. En este caso se puede decir que el método es convergente. A menos que se requiera un análisis cuidadoso, es muy fácil elegir un método que dé una sucesión de valores que poco a poco convergen a la solución de la ecuación dada [Burden *et al.*, 2002], [Nieves *et al.*, 2002], [Rodríguez, 2003], [Cordero *et al.*, 2006].

Resulta fácil mostrar gráficamente las circunstancias bajo las cuales el proceso converge. La figura 2.6a muestra una gráfica típica en la cual el proceso es convergente. Se traza la curva $y = g(x)$, y esto permite obtener de la gráfica el valor $y_0 = g(x_0)$. Entonces se requiere el valor $x_1 = y_0$, y esto se encuentra trazando la línea $y = x$. Así se puede obtener el valor apropiado de x. Trazando las líneas comenzando de x_0 a x_1, etc., se observa que los valores están convergiendo al punto x^*, donde $x^* = g(x^*)$, que es la solución de la ecuación. En este tipo de convergencia, la diferencia entre la aproximación y la solución verdadera siempre tienen el mismo signo; debido a esto, los valores se aproximan a la solución de manera estable. Este tipo de comportamiento se conoce como *monótonamente convergente*. La figura 2.6b muestra otro tipo de convergencia posible donde los valores oscilan en cualquier lado de la solución verdadera. Este tipo de *oscilación convergente* es muy conveniente debido a que los dos últimos valores dan límites en donde está la solución verdadera. Sin embargo, es igualmente posible que el método diverja; las figuras 2.6c y 2.6d muestran dos ejemplos sencillos de cómo puede ocurrir esto.

www.FreeLibros.me

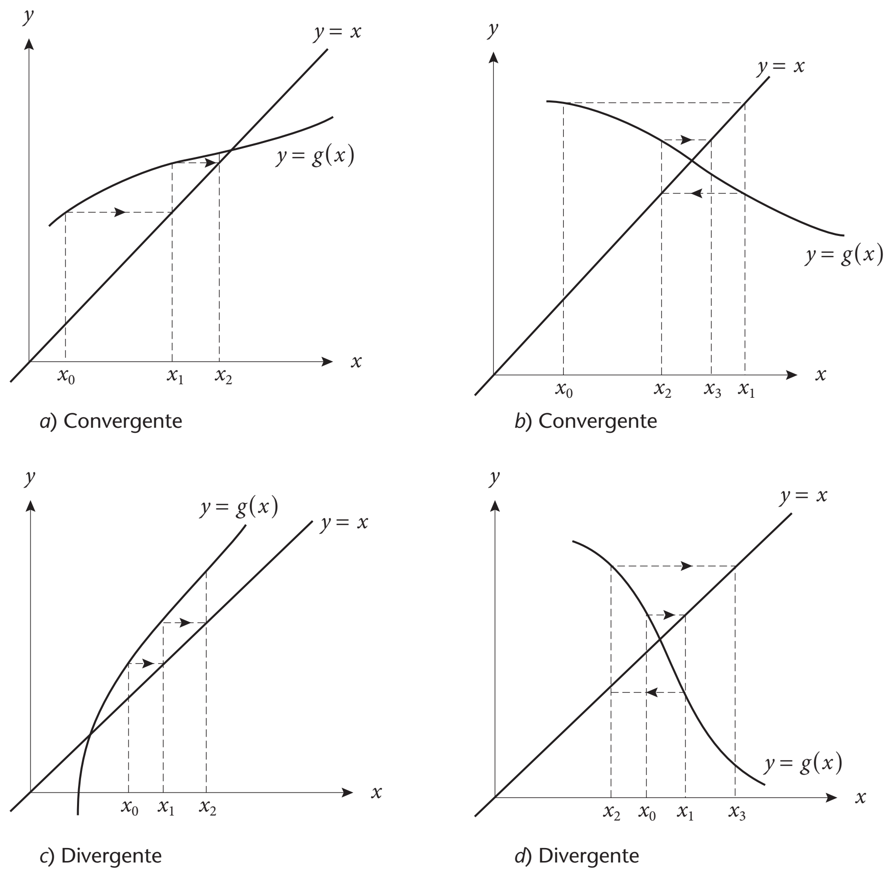


Figura 2.6 Método de punto fijo.

Del estudio de las gráficas queda claro que el proceso va a ser convergente cuando la derivada de la función $g(x)$ es menor que la derivada de la línea $y = x$. Por ejemplo, en la región cubierta por la iteración se requiere que

$$|g'(x)| < 1$$

Esto se puede demostrar teóricamente, si se tiene que

$$x_{n+1} = g(x_n), \ (n = 0, 1, 2, ...)$$

y la solución verdadera x^* satisface

$$x^* = g(x^*)$$

Restando estas ecuaciones se obtiene

$$x^* - x_{n+1} = g(x^*) - g(x_n)$$

Usando el teorema del valor medio en el lado derecho de la ecuación, se tiene

$$g(x^*) - g(x_n) = (x^* - x_n) g'(\zeta)$$

donde ζ es algún valor entre x^* y x_n. Si se define $\varepsilon_n = x^* - x_n$ la ecuación anterior será

$$\varepsilon_{n+1} = g'(\zeta) \cdot \ \varepsilon_n$$

www.FreeLibros.me

y, si $|g'(\zeta)|<1$, entonces el error siempre decrecerá paso a paso. Para $|g'(\zeta)|>1$, el error será creciente. En la sección 2.10.4 se proporciona el programa desarrollado en Matlab para este método. Para el método del punto fijo, se tiene el siguiente teorema.

Teorema 2.6 Sea g una función continua en el intervalo $[a,b]$ con $a \le g(x) \le b$ para $a \le x \le b$. Suponiendo que g' es continua en $[a,b]$ y que existe una constante $1>K>0$ tal que

$$|g'(x)| \le K \text{ para toda } x \text{ en } [a,b]$$

Si $g'(p) \neq 0$, entonces para cualquier $p_0 \in [a,b]$, la sucesión generada por

$$x_{n+1} = g(x_n), n = 1, 2, \ldots$$

converge linealmente a p en $[a,b]$, con $p = g(p)$.

EJEMPLO 2.4

Considerando la ecuación $f(x) = 1 + 2x - 3x^2 e^{-x} + 2x^3 \operatorname{sen}(x) e^{-\frac{x}{5}}$, calcular los cruces por cero dentro del intervalo $[4, 20]$; usar el método de punto fijo con un error máximo de 10^{-5}.

Solución. Para aplicar el método se iguala la función a cero y se despeja en variable en función de la misma variable; así se obtiene:

$$x_{n+1} = g(x_{n+1}) = \operatorname{sen}^{-1}\left(\frac{-1 - 2x_n + 3(x_n)^2 e^{-x_n}}{2(x_n)^3 e^{-\frac{x_n}{5}}} \right)$$

Como resultado de la aplicación de este programa se obtiene la tabla 2.4 que muestra todos los pasos intermedios hasta llegar a la convergencia.

Tabla 2.4a Tabla de valores que se obtienen al usar el método de punto fijo al calcular el primer cruce por cero.

n	x_n	$f(x_n)$	gx_{n+1}	$f(gx_{n+1})$
0	6.000000	–23.624104	6.185174	–0.306835
1	6.185174	–0.306835	6.187419	–0.003554
2	6.187419	–0.003554	6.187445	–4.11259e–5

Tabla 2.4b Tabla de valores que se obtienen al usar el método de punto fijo al calcular el segundo cruce por cero.

n	x_n	$f(x_n)$	gx_{n+1}	$f(gx_{n+1})$
0	9.000000	118.292948	9.503571	–0.208860
1	9.503571	–0.208860	9.502754	0.000241
2	9.502754	0.000241	9.502755	–2.80076e–7

www.FreeLibros.me

Tabla 2.4c Tabla de valores que se obtienen al usar el método de punto fijo al calcular tercer cruce por cero.

n	x_n	$f(x_n)$	gx_{n+1}	$f(gx_{n+1})$
0	12.000000	−143.229664	12.486554	0.419841
1	12.486554	0.419841	12.485240	−0.001255
2	12.485240	−0.001255	12.485244	3.75685e−6

Tabla 2.4d Tabla de valores que se obtienen al usar el método de punto fijo al calcular cuarto cruce por cero.

n	x_n	$f(x_n)$	gx_{n+1}	$f(gx_{n+1})$
0	15.000000	249.537288	15.800338	1.728367
1	15.800338	1.728367	15.805526	0.012094
2	15.805526	0.012094	15.805562	8.46963e−5

Tabla 2.4e Tabla de valores que se obtienen al usar el método de punto fijo al calcular el quinto cruce por cero.

n	x_n	$f(x_n)$	gx_{n+1}	$f(gx_{n+1})$
0	18.000000	−202.342578	18.733198	2.447108
1	18.733198	2.447108	18.725253	−0.028061
2	18.725253	−0.028061	18.725344	0.000321
3	18.725344	0.000321	18.725343	−3.68681e−6

2.6 Método de Newton-Raphson

Los métodos de cálculo estándar se utilizan para encontrar un método con rapidez de convergencia satisfactorio. Si la iteración ha alcanzado el punto x_n, entonces se requiere un incremento Δx_n que tomará el proceso hacia el punto solución x^*. Si se hace la expansión de $f(x^*)$ en series de Taylor se tiene que:

$$0 = f(x^*) \equiv f(x_n + \Delta x_n) = f(x_n) + \Delta x_n f'(x_n) + \frac{(\Delta x_n)^2}{2!} f''(x_n) + \cdots \tag{2.8}$$

Si la distancia Δx_n entre el punto de la iteración actual y la solución verdadera es suficientemente pequeña, entonces, tomando los dos primeros términos del lado derecho de la ecuación (2.8), se obtiene:

$$0 \approx f(x_n) + \Delta x_n f'(x_n)$$

Así, despejando Δx_n se llega a la expresión

$$\Delta x_n \approx -\frac{f(x_n)}{f'(x_n)}$$

Si se toma

$$\Delta x_n = x_{n+1} - x_n$$

despejando x_{n+1} y sustituyendo en el valor de Δx_n, se obtiene el *método de Newton-Raphson* como [Mathews, 2000], [Burden *et al.*, 2002], [Nieves *et al.*, 2002], [Rodríguez, 2003], [Cordero *et al.*, 2006]:

www.FreeLibros.me

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{2.9}$$

Si se traza la tangente a la curva en el punto x_0, entonces:

$$\tan\theta = \frac{f(x_0)}{-\Delta x} = f'(x_0)$$

Debido a esto, el paso Δx se encuentra gráficamente trazando la tangente a la curva en el punto de la presente iteración, y se encuentra donde corta el eje x. (Esto se muestra en forma gráfica en la figura 2.7.) Este valor de x en el cruce se usa en la siguiente iteración. La aproximación gráfica es útil para demostrar las propiedades de convergencia del método de Newton-Raphson.

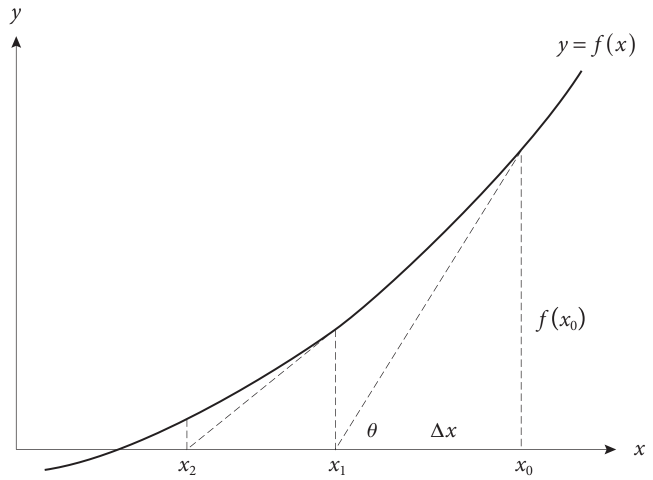


Figura 2.7 Método de Newton-Raphson.

Hay varias formas en las cuales *el método puede no converger*. La figura 2.8 muestra un ejemplo de una curva inclinada a lo largo del eje. Es evidente que el método de Newton-Raphson diverge en este caso. Como contraste, es claro que para una curva como la de la figura 2.7 el método de Newton-Raphson *converge monótonamente*.

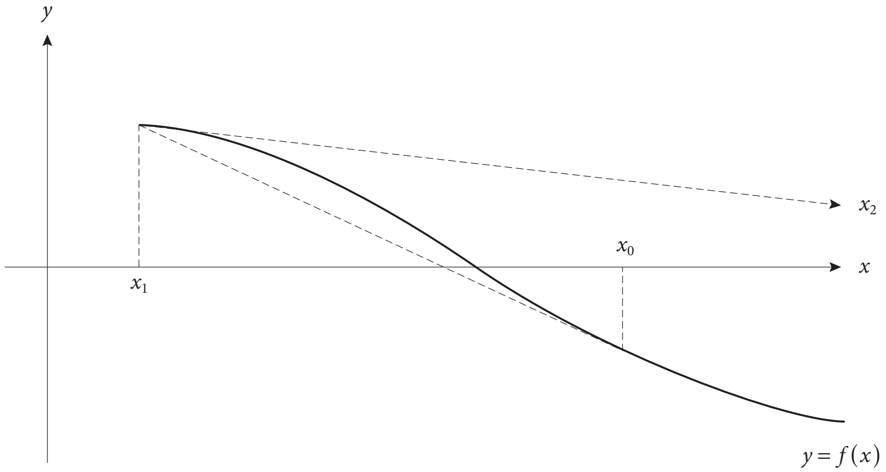


Figura 2.8 Divergencia del método de Newton-Raphson.

www.FreeLibros.me

Otra posibilidad es mostrarlo por las oscilaciones de la figura 2.9. Aquí, la derivada en el punto x_0 genera un punto x_1 en el que se calcula una derivada. Ésta genera un punto en la misma área que x_0 de tal forma que el proceso *oscila alrededor de un punto que no es* la solución. Por supuesto, es esencial en un proceso iterativo en computadora tener alguna forma de detenerlo si éste no converge.

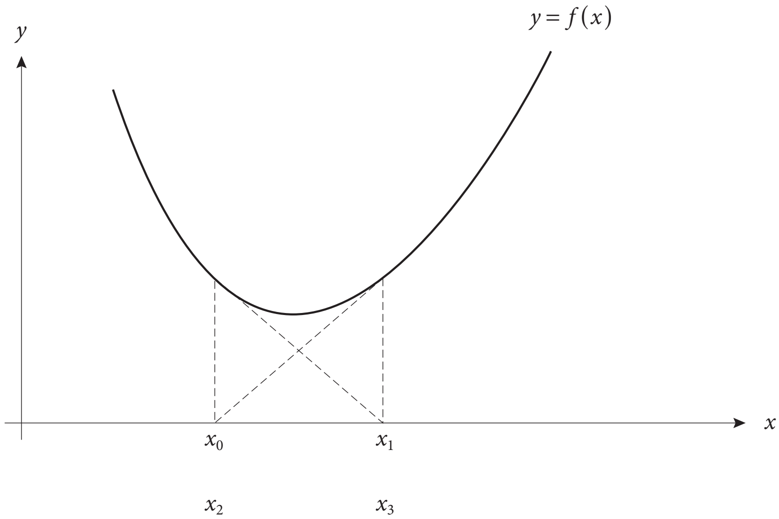


Figura 2.9 Oscilación del método de Newton-Raphson.

En vista de las dificultades que pueden surgir cuando se usa el método de Newton-Raphson, se necesita verificar con cuidado que el proceso converja satisfactoriamente. La convergencia se puede garantizar cuando la función tiene una segunda derivada que no cambia de signo en la región de iteración, es decir, satisface la siguiente condición

$$f(x)f''(x)>0$$

Si la segunda derivada se puede calcular fácilmente, es posible usar esta condición para verificar la convergencia. Si la convergencia no se puede calcular matemáticamente, entonces es necesario programar con cuidado para monitorear el progreso de las iteraciones y ver si están convergiendo. Si no hay signos de convergencia, se tendrá que cambiar el programa y emplear otro método iterativo que garantice la convergencia. Aun cuando el método muestre la convergencia, puede haber dificultades si la rapidez de convergencia es lenta.

Lo atractivo del método de Newton-Raphson es que, cuando los errores son pequeños, cada error es proporcional al cuadrado del error previo, lo cual ocasiona que la convergencia sea más rápida que la relación lineal que sostiene la iteración simple. El error relacionado con el método de Newton-Raphson se establece expandiendo $f(x)$ alrededor del punto x_n, así

$$0=f(x^*)=f(x_n)+(x^*-x_n)f'(x_n)+\frac{(x^*-x_n)^2}{2!}f''(\zeta) \tag{2.10}$$

donde ζ es algún valor de x entre x_n y x^*. El término $f(x_n)/f'(x_n)$ de la fórmula de Newton-Raphson se encuentra en la ecuación (2.9) y se sustituye en la (2.10) para dar

$$x_{n+1}=x_n+(x^*-x_n)+\frac{(x^*-x_n)^2}{2!}\frac{f''(\zeta)}{f'(x_n)}$$

De esta manera, probando que $f'(x_n)$ y $f''(\zeta)$ no son cero, el error es un múltiplo del cuadrado del error previo. Así se tiene que

www.FreeLibros.me

$$x^* - x_{n+1} = -\frac{(x^* - x_n)^2}{2!} \frac{f''(\zeta)}{f'(x_n)}$$

con lo que se establece el siguiente teorema.

Teorema 2.4 Suponiendo que f tiene segunda derivada continua y sea x^* tal que $f(x^*) = 0$ y $f'(x^*) \neq 0$. Si x_0 es lo suficientemente cercana a x^*, la sucesión $\{x_k\}_{k=0}^{\infty}$ generada por el método de Newton-Raphson converge a x^* con un orden de convergencia de al menos 2.

Este método se desarrolla en la plataforma que ofrece Matlab y se programa para dar todos los cruces por cero de una función no lineal que se mueve en plano real. El programa de cómputo se provee en la sección 2.10.5 de este capítulo. El resultado de ejecutar este programa se resume en la tabla 2.5, aquí se dan todos los pasos intermedios hasta lograr la convergencia.

EJEMPLO 2.5

Con la misma ecuación del ejemplo 2.1, es decir, $f(x) = 1 + 2x - 3x^2 e^{-x} + 2x^3 \operatorname{sen}(x) e^{-\frac{x}{5}}$. Calcular los cruces por cero dentro del intervalo $[4, 20]$; usar el método de Newton-Raphson con un error máximo de 10^{-5}.

Solución. La tabla 2.5 muestra los resultados numéricos del cálculo de los cruces por cero utilizando el método de Newton-Raphson.

Tabla 2.5a Tabla de valores que se obtienen al usar el método de Newton-Raphson al calcular el primer cruce por cero.

n	x_n	$f(x_n)$	$f'(x_n)$
0	6.000000	–23.624104	116.204971
1	6.203296	2.154944	136.658802
2	6.187528	0.011142	135.240326
3	6.187445	3.086892e–7	135.232833

Tabla 2.5b Tabla de valores que se obtienen al usar el método de Newton-Raphson al calcular el segundo cruce por cero.

n	x_n	$f(x_n)$	$f'(x_n)$
0	9.000000	118.292948	–204.321275
1	9.578955	–19.604657	–258.215107
2	9.503031	–0.070709	–256.082005
3	9.502755	–1.477762e–6	–256.071298

Tabla 2.5c Tabla de valores que se obtienen al usar el método de Newton-Raphson para calcular el tercer cruce por cero.

n	x_n	$f(x_n)$	$f'(x_n)$
0	12.000000	–143.229664	258.157006
1	12.554816	22.395130	323.174443
2	12.485518	0.087849	320.370373
3	12.485244	1.958645e–6	320.356083

www.FreeLibros.me

Tabla 2.5d Tabla de valores que se obtienen al usar el método de Newton-Raphson al calcular el cuarto cruce por cero.

n	x_n	$f(x_n)$	$f'(x_n)$
0	15.000000	249.537288	–253.302600
1	15.985135	–58.420480	–318.111897
2	15.801487	1.348182	–330.895919
3	15.805561	0.000322	–330.736660
4	15.805562	1.864464e–11	–330.736621

Tabla 2.5e Tabla de valores que se obtienen al usar el método de Newton-Raphson al calcular el quinto cruce por cero.

n	x_n	$f(x_n)$	$f'(x_n)$
0	18.000000	–202.342578	220.423607
1	18.917971	59.887064	308.343028
2	18.723749	–0.496616	311.469309
3	18.725343	1.849543e–5	311.492374

Análisis de resultados

La tabla 2.6 muestra una comparación de los cinco métodos; todos están implementados bajo las mismas restricciones. Analizando los resultados, se puede notar de manera simple que para converger a la tolerancia especificada, salvo el caso de bisección, el resto de los métodos convergen prácticamente en el mismo número de iteraciones.

Tabla 2.6 Tabla comparativa de número de iteraciones empleadas en el cálculo de cada cruce por cero, utilizando diferentes métodos.

Método utilizado	*Número de iteraciones para encontrar los cruces por cero con un error máximo de 10^{-5}*				
	Primero	*Segundo*	*Tercero*	*Cuarto*	*Quinto*
Bisección	20	19	16	19	19
Regla Falsa	7	3	3	3	3
Secante	4	3	3	3	3
Punto fijo	3	3	3	3	4
Newton-Raphson	3	3	3	4	3

Para el caso del ejemplo de aplicación, si se analiza la función a iterar en punto fijo, se tiene

$$x_{n+1} = \operatorname{sen}^{-1}\left(\frac{-1 - 2x_n + 3(x_n)^2 e^{-x_n}}{2(x_n)^3 e^{-\frac{x_n}{5}}} \right)$$

Si se sabe que el primer cruce por cero está en el intervalo $[6, 7]$; adicionalmente, si se toma como comienzo el inicio del intervalo, la función ángulo cuyo seno dará como resultado

www.FreeLibros.me

$$x_1 = \operatorname{sen}^{-1}\left(\frac{-1-2(6)+3(6)^2 e^{-6}}{2(6)^3 e^{-\frac{6}{5}}}\right) = -0.0980103287414036$$

Este resultado es el arco complementario de 2π. Así, el resultado correcto es

$$x_1 = 2\pi - 0.0980103287414036 = 6.18517497843818$$

Si se sabe de antemano que el seno es una función circular, en cada caso se debe hacer el ajuste para interpretar correctamente el resultado de la ecuación; si no, siempre se tendrá de forma errónea que la ecuación es no convergente.

2.7 Aproximaciones iniciales de los cruces por cero

Las siguientes ideas pueden ser útiles para calcular las aproximaciones iniciales de las soluciones de una ecuación:

La gráfica de la ecuación por resolver puede dar información para localizar los cruces por cero.

El conocimiento de las circunstancias físicas del problema que se está modelando pueden conducir a una buena aproximación inicial de los cruces por cero.

Alternativamente, en ocasiones es posible reescribir la ecuación en otra expresión equivalente y ésta nos puede indicar la posición aproximada de los cruces por cero.

En ocasiones la ecuación se pueden separar en dos partes, y la intersección de las gráficas de las dos funciones puede indicar con más claridad la ubicación de los cruces por cero que la gráfica de la función original.

Si se está considerando un programa automático de cómputo, entonces se puede hacer el cálculo sistemático de valores de la función hasta que se encuentren dos valores de signo contrario. Para funciones continuas, estos valores contienen un cruce por cero.

Cuando se tiene el caso de que en ciertas regiones parte de la ecuación es insignificante, es posible obtener una aproximación a la solución resolviendo la parte residual de la ecuación.

2.8 Sistemas de ecuaciones no lineales

Algunos de los métodos de la sección previa se pueden generalizar para obtener la solución de sistemas de ecuaciones no lineales, aunque el análisis de las propiedades de convergencia no es sencillo. Se analizan dos de ellos, como son el de Newton-Raphson y el de punto fijo multivariable. A continuación se proporciona una descripción de cada uno de ellos.

2.8.1 Newton-Raphson

En el caso del método de Newton-Raphson, éste se puede extender al caso de dos ecuaciones simultáneas no lineales con dos incógnitas [Grainger *et al.*, 1996], de la forma:

$$f(x,y)=0$$
$$g(x,y)=0$$

Si x_0, y_0 son la primera aproximación de la solución y Δx y Δy son los incrementos necesarios para alcanzar la solución correcta, entonces, expandiendo la función en series de Taylor se tiene

$$0 = f(x_0+\Delta x, y_0+\Delta y) = f(x_0,y_0) + \frac{\partial f}{\partial x}(x_0,y_0)\Delta x + \frac{\partial f}{\partial y}(x_0,y_0)\Delta y + \cdots$$

$$0 = g(x_0+\Delta x, y_0+\Delta y) = g(x_0,y_0) + \frac{\partial g}{\partial x}(x_0,y_0)\Delta x + \frac{\partial g}{\partial y}(x_0,y_0)\Delta y + \cdots$$

www.FreeLibros.me

Si se ignoran los términos de segundo orden y las derivadas parciales se designan como $\frac{\partial f}{\partial x}=a_{11}$, $\frac{\partial f}{\partial y}=a_{12}$, $\frac{\partial g}{\partial x}=a_{21}$ y $\frac{\partial g}{\partial y}=a_{22}$, se tiene el sistema de ecuaciones

$$a_{11}\Delta x_0 + a_{12}\Delta y_0 + f(x_0, y_0) = 0$$
$$a_{21}\Delta x_0 + a_{22}\Delta y_0 + g(x_0, y_0) = 0$$

Agrupando matricialmente en función de los incrementos Δx_0 y Δy_0 se obtiene:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} \Delta x_0 \\ \Delta y_0 \end{bmatrix} = -\begin{bmatrix} f(x_0, y_0) \\ g(x_0, y_0) \end{bmatrix}$$

Si se asigna la matriz de coeficientes a una nueva variable **A** y se premultiplica por la inversa, se llega a

$$\begin{bmatrix} \Delta x_0 \\ \Delta y_0 \end{bmatrix} = -\mathbf{A}^{-1}\begin{bmatrix} f(x_0, y_0) \\ g(x_0, y_0) \end{bmatrix}$$

Las nuevas aproximaciones son, por tanto:

$$x_1 = x_0 + \Delta x_0$$
$$y_1 = y_0 + \Delta y_0$$

Este proceso se puede repetir usando los valores de x_1 y y_1 del lado derecho y en la derivada parcial para encontrar nuevos valores. Así se obtiene el esquema iterativo

$$\begin{aligned} x_{r+1} &= x_r + \Delta x_r \\ & \qquad\qquad r = 0, 1, 2, \ldots \\ y_{r+1} &= y_r + \Delta y_r \end{aligned}$$

En términos generales se tienen n ecuaciones con n incógnitas de la forma

$$\mathbf{F}^{(r)} = \begin{bmatrix} f_1\left(x_1^{(r)}, x_2^{(r)}, \ldots, x_n^{(r)}\right) \\ f_2\left(x_1^{(r)}, x_2^{(r)}, \ldots, x_n^{(r)}\right) \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ f_n\left(x_1^{(r)}, x_2^{(r)}, \ldots, x_n^{(r)}\right) \end{bmatrix}$$

La matriz **A** se forma con los términos

$$a_{ij} = \frac{\partial f_i}{\partial x_i}$$

y los incrementos $\Delta\mathbf{x}^{(r)}$ se forman resolviendo sucesivamente el grupo de ecuaciones:

$$\Delta\mathbf{x}^{(r)} = -\mathbf{A}^{-1}\mathbf{F}^{(r)}, \qquad (r = 0, 1, 2, \ldots)$$

El programa de cómputo desarrollado en Matlab para este método se presenta en la sección 2.10.7 de este capítulo.

EJEMPLO 2.6

Con el método de Newton-Raphson resolver el siguiente sistema de ecuaciones sujeta a las siguientes condiciones iniciales, $x_0 = 1$, $y_0 = 1$ y $z_0 = 1$

www.FreeLibros.me

$$
\begin{aligned}
2x-3xy+2z^2&=1\\
x+7y+2yz&=2\\
3x+xy+8z&=3
\end{aligned}
$$

Solución. Aplicando el método de Newton-Raphson, el sistema iterativo queda de la siguiente manera

$$
\Delta\begin{bmatrix} x\\ y\\ z\end{bmatrix}^{(r)}=-\begin{bmatrix} 2-3y^{(r)} & -3x^{(r)} & 4z^{(r)}\\ 1 & 7+2z^{(r)} & 2y^{(r)}\\ 3+y^{(r)} & x^{(r)} & 8\end{bmatrix}^{-1}\begin{bmatrix} 2x^{(r)}-3x^{(r)}y^{(r)}+2\left(z^{(r)}\right)^2-1\\ x^{(r)}+7y^{(r)}+2y^{(r)}z^{(r)}-2\\ 3x^{(r)}+x^{(r)}y^{(r)}+8z^{(r)}-3\end{bmatrix}
$$

Sustituyendo las condiciones iniciales para obtener $\Delta\mathbf{x}^{(0)}$, se obtiene

$$
\Delta\begin{bmatrix} x\\ y\\ z\end{bmatrix}^{(0)}=-\begin{bmatrix} 2-3 & -3 & 4\\ 1 & 7+2 & 2\\ 3+1 & 1 & 8\end{bmatrix}^{-1}\begin{bmatrix} 2-3+2-1\\ 1+7+2-2\\ 3+1+8-3\end{bmatrix}=\begin{bmatrix} -0.7333\\ -0.6571\\ -0.6762\end{bmatrix}
$$

Numéricamente la primera iteración es

$$
\begin{aligned}
x_1&=x_0+\Delta x_0=1-0.7333=0.2667\\
y_1&=y_0+\Delta y_0=1-0.6571=0.3429\\
z_1&=z_0+\Delta z_0=1-0.6762=0.3238
\end{aligned}
$$

La tabla 2.7 muestra los resultados de las iteraciones hasta que el método converge.

Tabla 2.7 Método de Newton-Raphson.

Iteración	*x*	*y*	*z*
0	1.0000	1.0000	1.0000
1	0.2667	0.3429	0.3238
2	1.2794	0.1370	–0.1527
3	0.7739	0.1742	0.0656
4	0.6785	0.1834	0.1049
5	0.6746	0.1838	0.1065

Se dice que el método converge cuando cada valor de $\Delta\mathbf{x}$ es menor a una tolerancia especificada. Para el caso del ejemplo 2.6 se usó una tolerancia de 0.001.

2.8.2 Punto fijo multivariable

El método del punto fijo también se puede extender de manera simple para aplicarlo a un sistema de ecuaciones no lineales. El sistema de ecuaciones por iterar se obtiene reagrupando cada una de las ecuaciones para obtener un sistema de ecuaciones que separa en el lado izquierdo cada una de las variables; en el lado derecho se inserta una aproximación de cada variable, y así se calculan los nuevos valores. Éstos se usan para dar, a su vez, nuevos valores, y el proceso se repite en forma iterativa [Nieves *et al.*, 2002], [Burden *et al.*, 2002]. Si el método es adecuado, los valores se aproximan cada vez más a la solución verdadera. Si se parte del siguiente sistema de ecuaciones

www.FreeLibros.me

$$
\begin{aligned}
&f_0(x_0, x_1,\ldots, x_n)=0\\
&f_1(x_0, x_1,\ldots, x_n)=0\\
&\vdots\\
&f_n(x_0, x_1,\ldots, x_n)=0
\end{aligned}
$$

la fórmula general del método se obtiene reacomodando el sistema de ecuaciones como:

$$
\begin{aligned}
&x_0 = g(x_0, x_1,\ldots, x_n)\\
&x_1 = g(x_0, x_1,\ldots, x_n)\\
&\vdots\\
&x_n = g(x_0, x_1,\ldots, x_n)
\end{aligned}
$$

En forma iterativa, el sistema simplemente queda de la siguiente forma:

$$
\left.
\begin{aligned}
&x_0^{(r+1)} = g\left(x_0^{(r)}, x_1^{(r)},\ldots, x_n^{(r)}\right)\\
&x_1^{(r+1)} = g\left(x_0^{(r)}, x_1^{(r)},\ldots, x_n^{(r)}\right)\\
&\vdots\\
&x_n^{(r+1)} = g\left(x_0^{(r)}, x_1^{(r)},\ldots, x_n^{(r)}\right)
\end{aligned}
\right., \text{ con } r=0,1,\ldots,n \tag{2.11}
$$

La solución verdadera $\mathbf{x}^*$ satisface la ecuación,

$$
\mathbf{x}^* = \mathbf{G}(\mathbf{x}^*) \tag{2.12}
$$

Restando la ecuación (2.11) de la ecuación (2.12) se llega a,

$$
\mathbf{x}^* - \mathbf{x}^{r+1} = \mathbf{G}(\mathbf{x}^*) - \mathbf{G}(\mathbf{x}^r)
$$

Usando el teorema del valor medio en el lado derecho de la ecuación se tiene

$$
\mathbf{G}(\mathbf{x}^*) - \mathbf{G}(\mathbf{x}^r) = (\mathbf{x}^* - \mathbf{x}^r)\mathbf{G}'(\zeta)
$$

donde ζ es un vector de valores entre $\mathbf{x}^*$ y $\mathbf{x}^r$.

Si se define $\varepsilon^r = \mathbf{x}^* - \mathbf{x}^r$ la ecuación anterior será:

$$
\varepsilon^{r+1} = \mathbf{G}'(\zeta)\cdot\ \varepsilon^r
$$

y, si todos los valores de $|\mathbf{G}'(\zeta)|<1$, entonces el error en cada variable siempre decrecerá paso a paso. Para $|\mathbf{G}'(\zeta)|>1$ el error será creciente.

El programa desarrollado en Matlab para punto fijo multivariable con el desarrollo de Gauss y de Gauss-Seidel se provee en la sección 2.10.7 de este capítulo.

EJEMPLO 2.7

Para efectos de comparación, se aplica el método de punto fijo multivariable al sistema de ecuaciones del ejercicio 2.6, con las condiciones iniciales $x_0 = 1$, $y_0 = 1$ y $z_0 = 1$, es decir, el siguiente sistema de ecuaciones

$$
\begin{aligned}
2x - 3xy + 2z^2 &= 1\\
x + 7y + 2yz &= 2\\
3x + xy + 8z &= 3
\end{aligned}
$$

www.FreeLibros.me

Solución. Despejando de la primera ecuación la primera variable y así sucesivamente, el sistema iterativo tiene la siguiente estructura,

$$x_n = \frac{1+3x_{n-1}y_{n-1}-2z^2{}_{n-1}}{2}$$

$$y_n = \frac{2-x_{n-1}-2y_{n-1}z_{n-1}}{7}$$

$$z_n = \frac{3-3x_{n-1}-x_{n-1}y_{n-1}}{8}$$

Numéricamente, la primera iteración se obtiene sustituyendo en forma simultánea todas las condiciones iniciales en el sistema de ecuaciones anterior. Así se obtiene

$$x_1 = \frac{1+3x_0y_0-2z^2{}_0}{2} = \frac{1+3-2}{2} = 1$$

$$y_1 = \frac{2-x_0-2y_0z_0}{7} = \frac{2-1-2}{7} = -\frac{1}{7}$$

$$z_1 = \frac{3-3x_0-x_0y_0}{8} = \frac{3-3-1}{8} = -\frac{1}{8}$$

La forma de sustitución anterior se conoce con el nombre de *método de Gauss*. Una variante conocida con el nombre de "método de Gauss-Seidel" utiliza las soluciones que se van calculando en las siguientes ecuaciones; es decir, una vez que se calcula el valor de una variable, se utiliza inmediatamente en las ecuaciones sucesivas. Con este esquema, el sistema iterativo queda de la siguiente forma

$$x_n = \frac{1+3x_{n-1}y_{n-1}-2z^2{}_{n-1}}{2}$$

$$y_n = \frac{2-x_n-2y_{n-1}z_{n-1}}{7}$$

$$z_n = \frac{3-3x_n-x_ny_n}{8}$$

Con este método se obtienen los siguientes resultados en la primera iteración

$$x_1 = \frac{1+3x_0y_0-2z_0^2}{2} = \frac{1+3-2}{2} = 1$$

$$y_1 = \frac{2-x_1-2y_0z_0}{7} = \frac{2-1-2}{7} = -\frac{1}{7}$$

$$z_1 = \frac{3-3x_1-x_1y_1}{8} = \frac{3-3+\frac{1}{7}}{8} = \frac{1}{56}$$

La tabla 2.8 muestra los resultados utilizando ambos métodos de punto fijo multivariable. Comparando los resultados de ambos métodos, es decir, el método de Newton-Raphson (tabla 2.7) y el método de punto fijo multivariable con sus dos variantes (tabla 2.8), se puede deducir de manera simplista que el método de Newton-Raphson tiene mayor rapidez de convergencia; sin embargo, se debe tener en cuenta que el ejemplo de aplicación tiene derivada analítica, y esto simplifica la aplicación del método de Newton-

www.FreeLibros.me

Tabla 2.8 Método de punto fijo multivariable.

	Método de Gauss			*Método de Gauss-Seidel*		
Iteración	*x*	*y*	*z*	*x*	*y*	*z*
0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
1	1.0000	–0.1429	–0.1250	1.0000	–0.1429	0.0179
2	0.2701	0.1378	0.0179	0.2854	0.2457	0.2592
3	0.5555	0.2464	0.2691	0.5380	0.1907	0.1604
4	0.6329	0.1874	0.1496	0.6281	0.1872	0.1248
5	0.6556	0.1873	0.1228	0.6609	0.1846	0.1119
6	0.6691	0.1855	0.1138	0.6705	0.1840	0.1081
7	0.6732	0.1841	0.1086	0.6734	0.1838	0.1070
8	0.6741	0.1838	0.1071	0.6742	0.1838	0.1067
9	0.6744	0.1838	0.1067	0.6745	0.1838	0.1066
10	0.6745	0.1838	0.1066	0.6746	0.1838	0.1065
11	0.6746	0.1838	0.1066	0.6746	0.1838	0.1065
12	0.6746	0.1838	0.1065			
13	0.6746	0.1838	0.1065			

Raphson, además de no introducir errores causados por la implementación de un método de derivación numérica para el caso de problemas sin derivada analítica.

EJEMPLO 2.8

Aplicación del método de Newton-Raphson en la solución de flujos de potencia en un sistema eléctrico.

Solución. Usando el principio de suma de corrientes en un nodo igual a cero, es decir, el método de nodos, y de que a partir del estado estable de un sistema trifásico se puede representar por la secuencia positiva, se tendrá [Grainger *et al.*, 1996]

$$\sum_{i=1}^{n} I_i = I_1 + I_2 + ... + I_n = 0$$

donde n es el número de líneas conectadas al nodo. Para el caso de parámetros concentrados, se tiene que para el i-ésimo nodo la representación matemática lleva a;

$$\begin{aligned} I_1 &= Y_{i1}V_1 \\ I_2 &= Y_{i2}V_2 \\ &\vdots \\ I_n &= Y_{in}V_n \end{aligned}$$

donde $Y_{in} = \dfrac{1}{Z_{in}}$, con Z_{in} como la impedancia nodal de secuencia positiva. Sumando todas las corrientes y asignándole a la suma total una variable denotada como el i-ésimo nodo, se tiene

$$I_i = \sum_{k=1}^{n} Y_{ik} V_k$$

www.FreeLibros.me

Representando el método nodal en función de la potencia compleja

$$S_i^* = V_i^* I_i$$

Es decir,

$$S_i^* = V_i^* \sum_{k=1}^{n} Y_{ik} V_k$$

donde

$$S_i^* = P_i - jQ_i$$

$$V_i^* = |V_i|(\cos\delta_i - j\text{sen}\delta_i)$$

$$Y_{ik} = G_{ik} + jB_{ik} \quad \text{para} \quad k = 1, 2, \ldots, n$$

$$V_k = |V_k|(\cos\delta_k + j\text{sen}\delta_k) \quad \text{para} \quad k = 1, 2, \ldots, n$$

Sustituyendo, se obtiene la siguiente ecuación

$$P_i - jQ_i = |V_i|(\cos\delta_i - j\text{sen}\delta_i)\sum_{k=1}^{n}(G_{ik} + B_{ik})|V_k|(\cos\delta_k + j\text{sen}\delta_k)$$

Aplicando la ley distributiva y separando parte real e imaginaria se llega a

$$P_i = |V_i|\sum_{k=1}^{n}|V_k|G_{ik}(\cos\delta_i\cos\delta_k + j\text{sen}\delta_i\text{sen}\delta_k) + B_{ik}(\cos\delta_i\cos\delta_k + j\text{sen}\delta_i\text{sen}\delta_k)$$

$$Q_i = |V_i|\sum_{k=1}^{n}|V_k|G_{ik}(\cos\delta_i\cos\delta_k + j\text{sen}\delta_i\text{sen}\delta_k) + B_{ik}(\cos\delta_i\cos\delta_k + j\text{sen}\delta_i\text{sen}\delta_k)$$

La expansión de Taylor de P_i y Q_i alrededor de (δ_0, V_0) es:

$$P_i = P_i(\delta_0, V_0) + \Delta(\delta_0, V_0)\frac{\partial(P_i(\delta_0, V_0))}{\partial(\delta_0, V_0)} + \cdots + \frac{\Delta^n(\delta_0, V_0)}{n!}\frac{\partial(P_i(\delta_0, V_0))}{\partial(\delta_0, V_0)}$$

$$Q_i = P_i(\delta_0, V_0) + \Delta(\delta_0, V_0)\frac{\partial(Q_i(\delta_0, V_0))}{\partial(\delta_0, V_0)} + \cdots + \frac{\Delta^n(\delta_0, V_0)}{n!}\frac{\partial(Q_i(\delta_0, V_0))}{\partial(\delta_0, V_0)}$$

Tomando sólo dos términos de la expansión se obtiene en forma matricial

$$\begin{bmatrix} P_i(\delta_0, V_0) - P_i \\ Q_i(\delta_0, V_0) - Q_i \end{bmatrix} + \begin{bmatrix} \dfrac{\partial(P_i(\delta_0, V_0))}{\partial(\delta_0, V_0)} \\ \\ \dfrac{\partial(Q_i(\delta_0, V_0))}{\partial(\delta_0, V_0)} \end{bmatrix} [\Delta(\delta_0, V_0)] = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Si se sabe que P_i y Q_i son valores especificados y $P_i(\delta_0, V_0)$ y $Q_i(\delta_0, V_0)$ son valores calculados con condiciones iniciales de ángulo y voltaje nodales respecto a una referencia, denotados aquí por δ_0 y V_0, el primer término de la ecuación estará constituido por los incrementos o variaciones de P y Q, es decir, las diferencias entre los valores especificados en el nodo y los valores calculados. Esto se denota por

www.FreeLibros.me

$$\Delta P = P_i(\delta_0, V_0) - P_i$$

$$\Delta Q = Q_i(\delta_0, V_0) - Q_i$$

Así se obtiene la ecuación

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} + \begin{bmatrix} \dfrac{\partial(P_i(\delta_0, V_0))}{\partial(\delta_0, V_0)} \\ \\ \dfrac{\partial(Q_i(\delta_0, V_0))}{\partial(\delta_0, V_0)} \end{bmatrix} [\Delta(\delta_0, V_0)] = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

A la derivada parcial de $P_i(\delta_0, V_0)$ y $Q_i(\delta_0, V_0)$ respecto al ángulo y al voltaje modal se le conoce como ***jacobiano*** y lo denota con la ecuación:

$$\mathbf{J} = \begin{bmatrix} \partial P_i \\ \partial Q_i \end{bmatrix}$$

Así, sustituyendo, se obtiene finalmente

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} + \mathbf{J}[\Delta(\delta_0, V_0)] = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

La ecuación anterior tiene únicamente como incógnita los incrementos de ángulo y de voltaje nodal. Por tanto, despejando $[\Delta(\delta_0, V_0)]$, se obtiene

$$[\Delta(\delta_0, V_0)] = -\mathbf{J}^{-1} \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix}$$

Esta corrección se aplica de la siguiente manera

$$\delta_1 = \delta_0 + \Delta\delta_0$$
$$V_1 = V_0 + \Delta V_0$$

Con las ecuaciones previas se vuelven a calcular P_i y Q_i, así como sus incrementos; asimismo se calcula el jacobiano. Con la actualización se vuelven a calcular los incrementos de ángulo y de voltaje nodal. El proceso iterativo anterior se obtiene de acuerdo con una lógica propia, debido a que los criterios pueden ser variados; el criterio puede ser, por ejemplo, el número de iteraciones, o cuando los ΔP y ΔQ sean menores a una tolerancia.

Como caso de aplicación, se toma una red de 19 nodos, como se muestra en la figura 2.10. La red es trifásica con 6 puntos de generación y 13 puntos de carga. Esta red opera en forma balanceada; por tanto, se puede representar solamente con su modelo de secuencia positiva. Los parámetros de secuencia positiva de cada línea se muestran en la figura 2.11, así como los nodos de generación y de carga. A continuación se da una breve explicación de cada etapa.

Etapa 1

Se toma la red trifásica y se calculan los parámetros L, R, C y G. Se toma un caso base donde se tiene generación y carga en función de la potencia real P y la potencia reactiva Q. El esquema de conectividad se muestra en la figura 2.10.

Etapa 2

Una vez que se tienen los parámetros de las líneas, se aplica la transformación de componentes simétricas y se hace la representación con la secuencia positiva; esto es posible debido a que la red opera en estado estable. Los parámetros **Z**, **Y** de cada línea se muestran en la figura 2.11.

www.FreeLibros.me

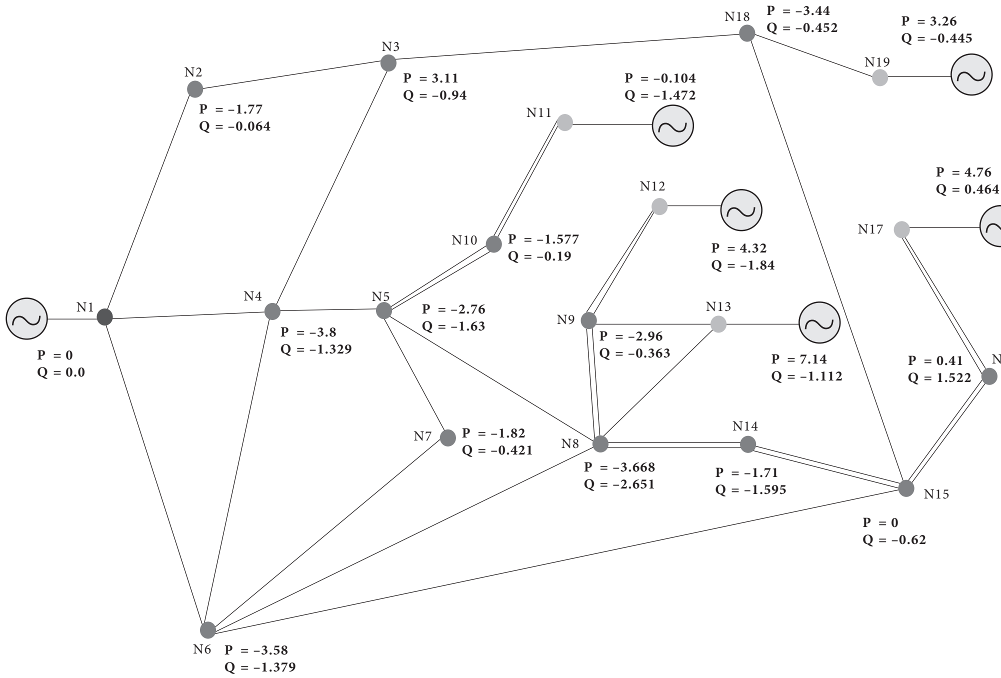
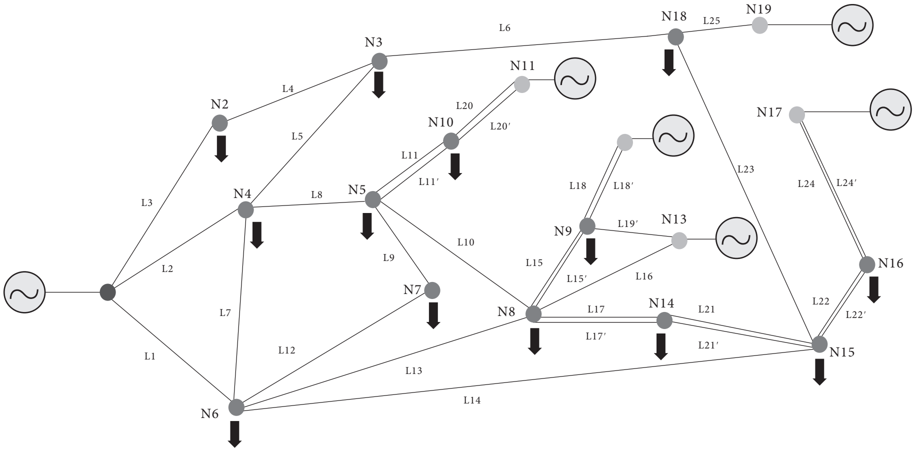


Figura 2.10 Datos de P y Q de la red trifásica.



L1 → Z= 0.0038+.04938j, Y=1.4978j	L11 → Z= 0.0035+.04520j, Y=1.2759j	L19 → Z= 0.0027+.03326j, Y=0.9817j
L2 → Z= 0.0033+.04393j, Y=1.1583j	L11′ → Z= 0.0043+.04540j, Y=1.2604j	L20 → Z= 0.0006+.00820j, Y=0.2285j=L20′
L3 → Z= 0.0002+.00240j, Y=0.0672j	L12 → Z= 0.0007+.00894j, Y=0.2638j	L21 → Z= 0.0040+.05347j, Y=1.5065j=L21′
L4 → Z= 0.0033+.04289j, Y=1.2143j	L13 → Z= 0.0035+.04341j, Y=1.2800j	L22 → Z= 0.0003+.00328j, Y=0.0967j
L5 → Z= 0.0019+.02057j, Y=0.5658j	L14 → Z= 0.0041+.05013j, Y=0.4796j	L22′ → Z= 0.0002+.00238j, Y=0.0703j
L6 → Z= 0.0019+.02562j, Y=0.7199j	L15 → Z= 0.0006+.00707j, Y=0.2088j	L23 → Z= 0.0022+.02729j, Y=0.8056j
L7 → Z= 0.0007+.00839j, Y=0.2348j	L15′ → Z= 0.0040+.05309j, Y=1.4954j	L24 → Z= 0.0011+.01750j, Y=0.6367j
L8 → Z= 0.0011+.01374j, Y=0.4281j	L16 → Z= 0.0029+.03539j, Y=1.0444j	L24′ → Z= 0.0014+.01820j, Y=0.5094j
L9 → Z= 0.0009+.01201j, Y=0.3544j	L17 → Z= 0.0022+.02890j, Y=0.7967j= L1′	L25 → Z= 0.0044+.05680j, Y=1.6094j
L10 → Z= 0.0035+.04510j, Y=1.2953j	L18 → Z= 0.0040+.05331j, Y=1.4954j=L18′	

Figura 2.11 Datos de las líneas de la red trifásica.

Etapa 3

Con estos datos se programan flujos de potencia. Como resultado de la simulación, se obtienen los voltajes nodales y los ángulos referidos al nodo *Slack* el cual se deja con el ángulo igual a cero. La figura 2.12 muestra el resultado de la simulación.

www.FreeLibros.me

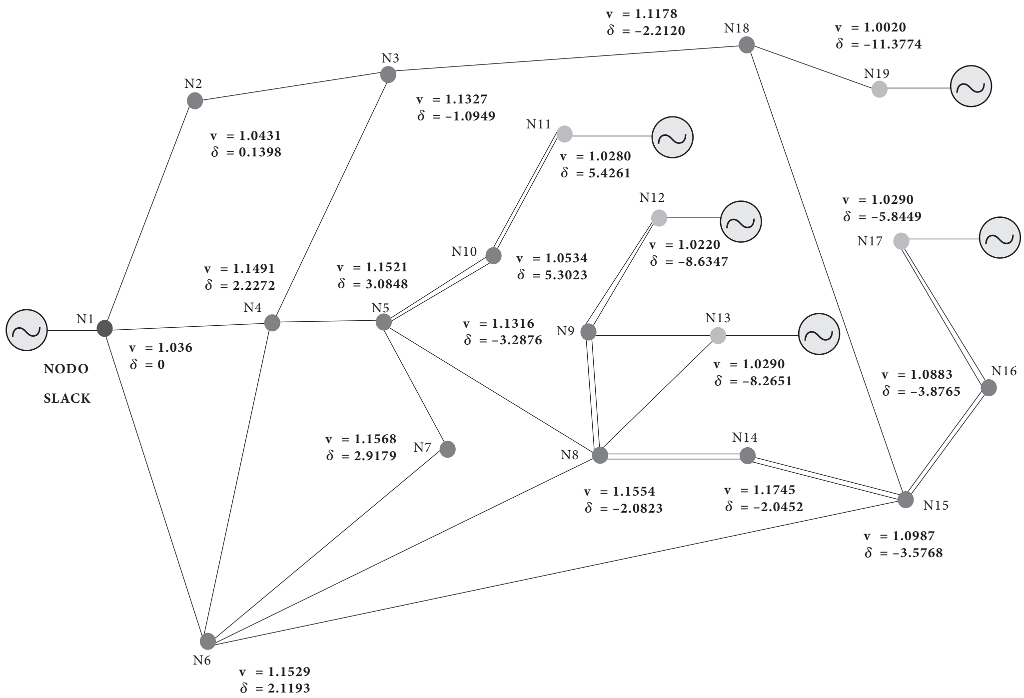


Figura 2.12 Voltajes y ángulos nodales calculados.

Etapa 4

Con los datos obtenidos se calcula el flujo de potencia de cada línea y su dirección. Esto se muestra en la figura 2.13.

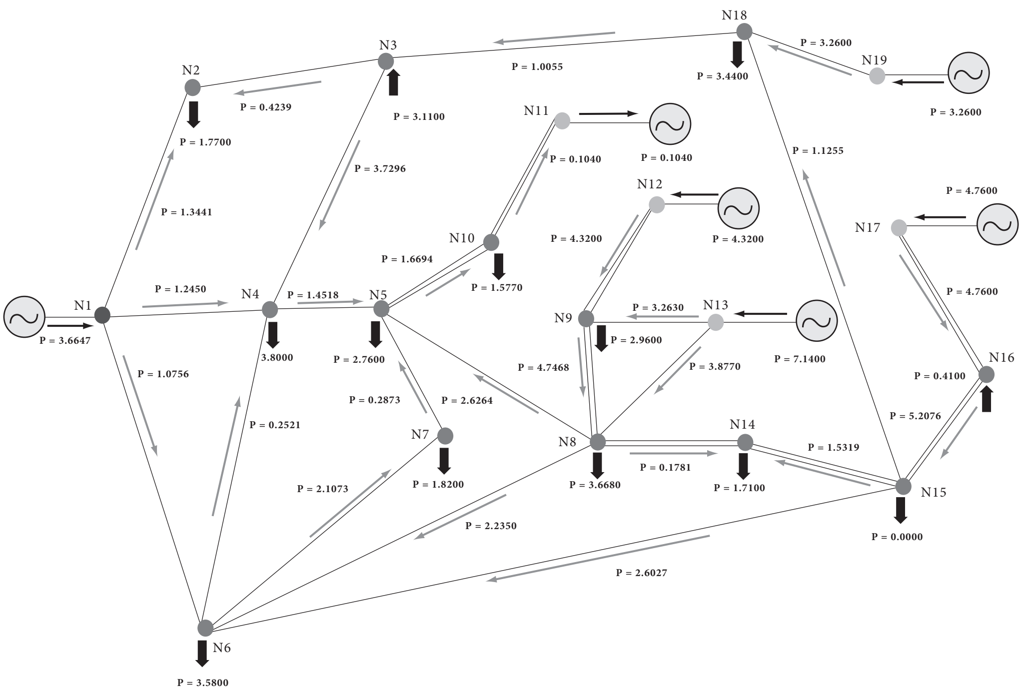


Figura 2.13 Flujo por cada línea.

www.FreeLibros.me

El problema de flujos de potencia consiste en determinar en una red eléctrica, la dirección y la cantidad de potencia que circula por cada línea de transmisión. Este problema se resuelve con la técnica de Newton-Raphson, debido a que es un problema no lineal. Como el punto de convergencia en un sistema no lineal depende de las condiciones iniciales, para garantizar la convergencia en un punto que físicamente sea realizable o posible, se dan condiciones iniciales muy cercanas a los valores nominales de la red eléctrica.

2.9 Comparación de métodos

El método más sencillo es el de bisección, que siempre es convergente. Tiene la desventaja de que su velocidad de convergencia es baja. Es totalmente conveniente como un método preliminar para hacer aproximaciones burdas de las soluciones, las cuales se pueden después refinar con métodos más perfeccionados. Otro método siempre convergente es el de la falsa posición. En ciertas circunstancias, este método es equivalente al de la secante y, entonces, tiene una buena rapidez de convergencia. Sin embargo, la figura 2.14 muestra una situación en la cual la convergencia del método de falsa posición puede resultar lentísima. Cuando se satisfacen las condiciones para la convergencia, se deben usar métodos como el de la secante o el método de Newton-Raphson.

Para un programa de computación automático, es siempre preferible una combinación de un método siempre convergente, como el de la falsa posición, con uno de convergencia rápida como el de Newton-Raphson. El método de falsa posición define un intervalo que contiene la solución, y el valor dado por el método de Newton-Raphson se acepta como el punto de la nueva iteración si está dentro de este intervalo. Si no, el valor dado por el método de falsa posición se usa como el punto de la nueva iteración. Con esto se mantienen dos puntos de lados opuestos del cruce por cero, y el método de Newton-Raphson se usará solamente cuando se obtenga un valor dentro de este rango.

Todos estos métodos se pueden aplicar para encontrar las raíces de un polinomio; sin embargo, sólo se podrán determinar las raíces reales y simples; pero si no se tienen la experiencia y el cuidado adecuados, se tendrá divergencia aun cuando en apariencia se está implementando un método cuya prueba de convergencia es positiva. Como un ejemplo de ello se prueba la función simple dada por:

$$f(x) = x^2 + 2x + 10$$

Si se aplica el método de punto fijo a esta ecuación, despejando de la parte lineal se obtiene:

$$g(x) = -\tfrac{1}{2}\left(x^2 + 10\right)$$

La derivada de $g(x)$ es,

$$g'(x) = -x$$

De acuerdo con el teorema 2.6, con la condición inicial dentro del intervalo $[-1, +1]$, se tendría un esquema convergente. La tabla 2.9 muestra las sucesiones de resultados generadas cuando se tienen diferentes puntos de inicio dentro del intervalo.

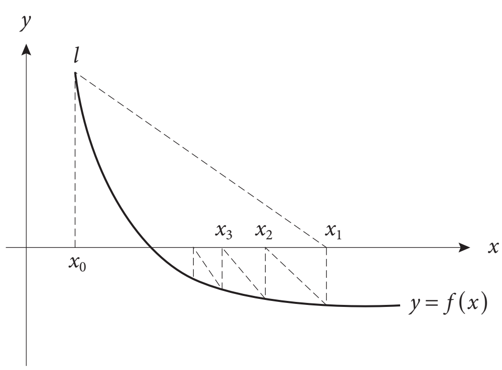


Figura 2.14 Convergencia lenta del método de la falsa posición.

www.FreeLibros.me

Tabla 2.9 Resultados de aplicar el método de punto fijo a un polinomio de grado 2.

Iteración	*Condición inicial*				
	$x_0 = -1$	$x_0 = \frac{1}{2}$	$x_0 = 0$	$x_0 = \frac{1}{2}$	$x_0 = 1$
1	−5.5000	−5.1250	−5.0000	−5.1250	−5.5000
2	−20.1250	−18.1328	−17.5000	−18.1328	−20.1250
3	−207.5078	−169.3994	−158.1250	−169.3994	−207.5078
4	−2.1535e4	−1.4353e4	−1.2507e4	−1.4353e4	−2.1535e4
5	−2.3187e8	−1.0301e8	−7.8210e7	−1.0301e8	−2.3187e8
6	−2.6882e16	−5.3051e15	−3.0584e15	−5.3051e15	−2.6882e16
7	−3.6133e32	−1.4072e31	−4.6768e30	−1.4072e31	−3.6133e32

Analizando la tabla anterior se puede notar que, a pesar de la prueba de convergencia positiva, en realidad el método es *incondicionalmente inestable*. La razón por la cual la solución es incondicionalmente inestable es que el polinomio tiene raíces en el plano complejo, como es el caso normal en un polinomio, y los métodos descritos en este capítulo trabajan sólo en el plano real. Por esta razón el uso de este tipo de métodos para encontrar las n raíces de un polinomio de orden n-ésimo es muy limitado.

Si se utilizan los métodos de la secante, de punto fijo o de Newton-Raphson para obtener las raíces de un polinomio n-ésimo, se restringen al cálculo de las raíces reales. Se debe, por supuesto, tener la certeza absoluta de que estas soluciones existen o los métodos divergen. Adicionalmente, se deben evitar los puntos de inconsistencia, es decir, puntos donde la función está en un mínimo local. Por otro lado, en el caso de los métodos de bisección y regla falsa aplicados al problema de cálculo de raíces, si se tienen raíces de multiplicidad par, entonces la función no tendrá cambios de signo en la cercanía de la raíz y, por tanto, estos métodos no encuentran dicha raíz.

Cuando se tienen raíces de multiplicidad par, hay siempre una región en la cual $f(x)f''(x) > 0$, de tal manera que el método de Newton-Raphson se puede usar garantizando convergencia sólo si se tiene una aproximación inicial suficientemente cercana.

Los métodos adecuados para la solución del problema de cálculo de raíces se analizan en el capítulo 3 de este mismo texto.

2.10 Programas desarrollados en Matlab

Esta sección proporciona los códigos de los programas desarrollados en Matlab para todos los ejercicios propuestos. A continuación se da la lista de ellos:

2.10.1 Método de bisección
2.10.2 Método de falsa posición o regla falsa
2.10.3 Método de la secante
2.10.4 Método de punto fijo
2.10.5 Método de Newton-Raphson
2.10.6 Método de Newton-Raphson para sistemas de ecuaciones
2.10.7 Método de punto fijo multivariable; Gauss y Gauss-Seidel

2.10.1 Método de bisección

El método de bisección se basa en dividir el intervalo en dos partes y verificar en cuál subintervalo se encuentra el cruce por cero. El método da como resultado un valor para la variable independiente dentro de un intervalo, para el cual la función evaluada da cero o es menor a una tolerancia especificada. En Matlab, cuando se usa el signo % significa que se está haciendo un comentario. Adicionalmente, las funciones propias de Matlab aparecen sombreadas.

www.FreeLibros.me

Programa principal del método de bisección

```
%  Programa principal para determinar los cruces por cero de una función continua
clear all
clc
%  Utilizando el método de bisección, determinar los cruces por cero de la función.
%  fx = 1 + 2.*x - 3.*x.^2.*exp(-x) + 2.*x.^3.*sin(x).*exp(-x./5) dentro del intervalo
%  [4,20]. Utilizar un error relativo de 1e-6.

r = 0;  % Inicia el contador de cruces por cero
%  Ciclo para calcular todos los cruces por cero dentro del intervalo [4,20]
for k = 4:20
    l = k+1;
    % Evaluación de la función en el punto k.
    gk = 1 + 2.*k - 3.*k.^2.*exp(-k) + 2.*k.^3.*sin(k).*exp(-k./5);
    % Evaluación de la función en el punto l=k+1.
    gl = 1 + 2.*l - 3.*l.^2.*exp(-l) + 2.*l.^3.*sin(l).*exp(-l./5);
    % Condicional que marca el subintervalo donde se encuentra un cruce por cero.
    if  gk*gl  <  1e-3
        % Método de bisección.
        [Cero,Mat]=Biseccion(k,l);
        r = r + 1;                 % Contador de cruces por cero.
        Cruce(r) = Cero;           % Almacena los cruces por cero.
        dr = length(Mat(:,1));     % Número de iteraciones para encontrar el cruce por
                                   % cero.
        dc = length(Mat(1,:));     % Número de variables por almacenar.
        M(r,1:dr,1:dc) = Mat;      % Matriz que almacena todas las iteraciones de todos
                                   % los cruces por cero.
    end
end
M1(:,:) = M(1,:,:);               % Iteraciones del primer cruce por cero.
M2(:,:) = M(2,:,:);               % Iteraciones del segundo cruce por cero.
M3(:,:) = M(3,:,:);               % Iteraciones del tercer cruce por cero.
M4(:,:) = M(4,:,:);               % Iteraciones del cuarto cruce por cero.
M5(:,:) = M(5,:,:);               % Iteraciones del quinto cruce por cero.
```

Función de Matlab llamada bisección

```
%  Función del método de Bisección para cálculo de cruces por cero.
%  de una función no lineal que se mueve en plano real.
%
%  El método calcula un cruce por cero.
%
%  La función se llama de la siguiente manera:
%
%              [Cero,Mat]=Biseccion(a,b)
%
%  Entradas:
%              a  -- Límite inferior del intervalo.
%              b  -- Límite superior del intervalo.
%
%  Salida:
%              Cero -- Valor de la variable para la cual la magnitud de la función es
%              cero o menor a una tolerancia especificada previamente.
%              Mat  -- Vector que contiene todas las iteraciones hasta converger.
%
function[Cero,Mat] = Biseccion(a,b);
Err = 1;       % Inicializa el error para ingresar al ciclo iterativo.
tol = 1e-8;    % Tolerancia especificada para la convergencia.
c   = 0;       % Inicializa el contador de iteraciones.
while Err > tol & c < 30
   % Valor de la función al inicio del intervalo
   fa = 1 + 2.*a - 3.*a.^2.*exp(-a) + 2.*a.^3.*sin(a).*exp(-a./5);
   % Valor de la función al final del intervalo
   fb = 1 + 2.*b - 3.*b.^2.*exp(-b) + 2.*b.^3.*sin(b).*exp(-b./5);
```

www.FreeLibros.me

```
    % Cálculo del punto medio.
    h  = (a+b)/2;
    % Valor de la función en el punto medio
    fh = 1 + 2.*h - 3.*h.^2.*exp(-h) + 2.*h.^3.*sin(h).*exp(-h./5);
    % Contador de iteraciones para no dejar en un ciclo el programa en caso de alguna
    % inconsistencia.
    c = c + 1;
    % Matriz que almacena los resultados de cada iteración.
    Mat(c,:) = [a fa b fb h fh];
    % Discriminante para determinar el nuevo intervalo.
    disc = fh*fa;
    % El cruce por cero cumple con el criterio de error.
    if abs(disc) <= tol
        Err  = 0;
        Cero = h;
    % Definición del nuevo intervalo al no cumplir el criterio de error.
    elseif disc > tol
        a = h;
        b = b;
    % Definición del nuevo intervalo al no cumplir el criterio de error.
    elseif disc < tol
        a = a;
        b = h;
    end
    Err = abs(disc);    % Criterio de error.
end
% Cruce por cero que determina el método de bisección.
Cero = h;
```

2.10.2 Método de regla falsa o falsa posición

El método de regla falsa se basa en trazar una línea recta entre dos puntos cuya función evaluada tiene signo contrario. El método encuentra el cruce entre la línea recta y el eje *x*, evalúa la función en este punto y verifica en cuál subintervalo se encuentra el cruce por cero. Realiza esta operación en forma iterativa hasta la convergencia. En Matlab, cuando se usa el signo % significa que se está haciendo un comentario. Adicionalmente las funciones propias de Matlab aparecen sombreadas.

Programa principal del método de regla falsa o falsa posición

```
%  Programa principal para determinar los cruces por cero de una función continua
clear all
clc
%  Utilizando el método de regla falsa, determinar los cruces por cero de la función.
%  fx = 1 + 2.*x - 3.*x.^2.*exp(-x) + 2.*x.^3.*sin(x).*exp(-x./5) dentro del
%  intervalo [4,20]. Utilizar un error relativo de 1e-6.
r=0;    % Inicia contador de cruces por cero.
% Ciclo para calcular todos los cruces por cero dentro del intervalo [4,20].
for k = 4:20
    l = k+1;
    % Evaluación de la función en el punto k.
    gk = 1 + 2.*k - 3.*k.^2.*exp(-k) + 2.*k.^3.*sin(k).*exp(-k./5);
    % Evaluación de la función en el punto l=k+1.
    gl = 1 + 2.*l - 3.*l.^2.*exp(-l) + 2.*l.^3.*sin(l).*exp(-l./5);
    % Condicional que marca el subintervalo donde se encuentra un cruce por cero.
    if  gk*gl  <  1e-3
        % Método de regla falsa o falsa posición.
        [Cero,Mat] = ReglaFalsa(k,l);
        r = r + 1;                % Contador de cruces por cero.
        Cruce(r) = Cero;          % Almacena los cruces por cero.
        dr = length(Mat(:,1));    % Número de iteraciones para encontrar el cruce por
                                  % cero.
        dc = length(Mat(1,:));    % Número de variables por almacenar.
```

www.FreeLibros.me

```
        M(r,1:dr,1:dc) = Mat;     % Matriz que almacena todas las iteraciones de
                                  % todos los cruces por cero.
    end
end
M1(:,:) = M(1,:,:);    % Iteraciones del primer cruce por cero.
M2(:,:) = M(2,:,:);    % Iteraciones del segundo cruce por cero.
M3(:,:) = M(3,:,:);    % Iteraciones del tercer cruce por cero.
M4(:,:) = M(4,:,:);    % Iteraciones del cuarto cruce por cero.
M5(:,:) = M(5,:,:);    % Iteraciones del quinto cruce por cero.
```

Función de Matlab llamada regla falsa

```
%  Función del método de Regla Falsa o Falsa Posición para cálculo de cruces por cero
%  de una función no lineal que se mueve en plano real.
%
%  El método calcula un cruce por cero.
%
%  La función se llama de la siguiente manera.
%
%           [Cero,Mat]=ReglaFalsa(a,b).
%
%  Entradas:
%           a  -- Límite inferior del intervalo.
%           b  -- Límite superior del intervalo.
%
%  Salida:
%           Cero -- Valor de la variable para la cual la magnitud de la función es
%           cero o menor a una tolerancia especificada previamente.
%           Mat  -- Vector que contiene todas las iteraciones hasta converger.
%
function[Cero,Mat] = ReglaFalsa(a,b);
Err = 1;    % Inicializa el error para ingresar al ciclo iterativo.
tol = 1e-5; % Tolerancia especificada para la convergencia.
c   = 0;    % Inicializa el contador de iteraciones.
while Err > tol & c < 30
   % Valor de la función al inicio del intervalo.
   fa = 1 + 2.*a - 3.*a.^2.*exp(-a) + 2.*a.^3.*sin(a).*exp(-a./5);
   % Valor de la función al final del intervalo.
   fb = 1 + 2.*b - 3.*b.^2.*exp(-b) + 2.*b.^3.*sin(b).*exp(-b./5);
   % Punto de cruce de la recta entre los puntos [a,f(a)] y [b,f(b)].
   h  = a - fa*(b-a)/(fb-fa);
   % Valor de la función en el punto de cruce.
   fh = 1 + 2.*h - 3.*h.^2.*exp(-h) + 2.*h.^3.*sin(h).*exp(-h./5);
   % Contador de iteraciones para no dejar en un ciclo el programa en caso de alguna
   % inconsistencia.
   c = c + 1;
   % Matriz que almacena los resultados de cada iteración.
   Mat(c,:) = [a fa b fb h fh];
   % Discriminante para determinar el nuevo intervalo.
   disc = fh*fa;
   % El cruce por cero cumple con el criterio de error.
   if abs(disc) <= tol
       Err  = 0;
       Cero = h;
   % Definición del nuevo intervalo al no cumplir el criterio de error.
   elseif disc > tol
       a = h;
       b = b;
   % Definición del nuevo intervalo al no cumplir el criterio de error.
   elseif disc < tol
       a = a;
       b = h;
   end
   Err = abs(disc);   % Criterio de error.
end
```

www.FreeLibros.me

```
% Cruce por cero que determina el método de regla falsa o falsa posición.
Cero = h;
```

2.10.3 Método de la secante

El método de la secante es similar al método de regla falsa, sólo que utiliza los puntos en sucesión estricta Por esta razón, es necesario hacer la prueba de convergencia antes de su implementación, o bien, estar verificando que las iteraciones no diverjan. En Matlab, cuando se usa el signo % significa que se está haciendo un comentario. Adicionalmente, las funciones propias de Matlab aparecen sombreadas.

Programa principal del método de la secante

```
%  Programa principal para determinar los cruces por cero de una función continua
clear all
clc
%  Utilizando el método de la secante, determinar los cruces por cero de la función.
%  fx = 1 + 2.*x - 3.*x.^2.*exp(-x) + 2.*x.^3.*sin(x).*exp(-x./5) dentro del
%  intervalo [4,20]. Utilizar un error relativo de 1e-6.
r=0;   % Inicia el contador de cruces por cero.
%  Ciclo para calcular todos los cruces por cero dentro del intervalo [4,20].
for k = 4:20
    l = k+1;
    % Evaluación de la función en el punto k.
    gk = 1 + 2.*k - 3.*k.^2.*exp(-k) + 2.*k.^3.*sin(k).*exp(-k./5);
    % Evaluación de la función en el punto l=k+1.
    gl = 1 + 2.*l - 3.*l.^2.*exp(-l) + 2.*l.^3.*sin(l).*exp(-l./5);
    % Condicional que marca el subintervalo donde se encuentra un cruce por cero.
    if gk*gl < 1e-3
        % Método de la secante.
        [Cero,Mat] = Secante(k,l);
        r = r + 1;                 % Contador de cruces por cero.
        Cruce(r) = Cero;           % Almacena los cruces por cero.
        dr = length(Mat(:,1));     % Número de iteraciones para encontrar el cruce por
                                   % cero.
        dc = length(Mat(1,:));     % Número de variables a almacenar.
        M(r,1:dr,1:dc) = Mat;      % Matriz que almacena todas las iteraciones de
                                   % todos los cruces por cero.
    end
end
M1(:,:) = M(1,:,:);    % Iteraciones del primer cruce por cero.
M2(:,:) = M(2,:,:);    % Iteraciones del segundo cruce por cero.
M3(:,:) = M(3,:,:);    % Iteraciones del tercer cruce por cero.
M4(:,:) = M(4,:,:);    % Iteraciones del cuarto cruce por cero.
M5(:,:) = M(5,:,:);    % Iteraciones del quinto cruce por cero.
```

Función de Matlab llamada secante

```
%  Función del método de la secante para cálculo de cruces por cero de una función no
%  lineal que se mueve en plano real.
%  El método calcula un cruce por cero.
%  La función se llama de la siguiente manera.
%              [Cero,Mat]=Secante(a,b).
%
%  Entradas:
%              a  -- Límite inferior del intervalo.
%              b  -- Límite superior del intervalo.
%
%  Salida:
%              Cero -- Valor de la variable para la cual la magnitud de la función es
%              cero o menor a una tolerancia especificada previamente.
%              Mat  -- Vector que contiene todas las iteraciones hasta convergencia.
%
```

www.FreeLibros.me

```
function[Cero,Mat] = Secante(a, b);
Err = 1;           % Inicializa el error para ingresar al ciclo iterativo.
tol = 1e-6;        % Tolerancia especificada para la convergencia.
%  Valor de la función al inicio del intervalo.
fa = 1 + 2.*a - 3.*a.^2.*exp(-a) + 2.*a.^3.*sin(a).*exp(-a./5);
%  Valor de la función al final del intervalo.
fb = 1 + 2.*b - 3.*b.^2.*exp(-b) + 2.*b.^3.*sin(b).*exp(-b./5);
%  Inicializa la matriz para almacenar datos generados.
Mat = [a fa
       b fb];
%  Contador de iteraciones, inicia en dos pues de antemano se tienen dos cálculos.
c   = 2;
while Err > tol & c < 20
   % Punto de cruce de la recta secante con el eje x.
   h  = a - fa*(b-a)/(fb-fa);
   % Valor de la función en el punto de cruce.
   fh = 1 + 2.*h - 3.*h.^2.*exp(-h) + 2.*h.^3.*sin(h).*exp(-h./5);
   % Contador de iteraciones para no dejar en un ciclo el programa en caso de alguna
   % inconsistencia.
   c = c + 1;
   % Matriz que almacena los resultados de cada iteración.
   Mat(c,:) = [h fh];
   % Los nuevos puntos para la siguiente iteración son:
   a = b;   fa = fb;
   b = h;   fb = fh;
   % Criterio de error.
   Err = abs(fh);
end
%  Cruce por cero que determina el método de la secante.
Cero = h;
```

2.10.4 Método de punto fijo

El método de punto fijo se basa en despejar la variable de la función, de esa forma la variable queda en función de sí misma. Debido a esto se necesita una prueba de convergencia para garantizar que desde el punto inicial se converge a la solución deseada. En Matlab, cuando se usa el signo % significa que se está haciendo un comentario; adicionalmente, las funciones propias de Matlab aparecen sombreadas.

Programa principal del método de punto fijo

```
%  Programa principal para determinar los cruces por cero de una función continua
clear all
clc
%  Utilizando el método de Punto fijo, determinar los cruces por cero de la función
%  fx = 1 + 2.*x - 3.*x.^2.*exp(-x) + 2.*x.^3.*sin(x).*exp(-x./5) dentro del
%  intervalo [4,20]. Utilizar un error relativo de 1e-6.
r=0;     % Inicia contador de cruces por cero.
% Ciclo para calcular todos los cruces por cero dentro del intervalo [4,20].
for k = 4:20
    l = k+1;
    % Evaluación de la función en el punto k.
    gk = 1 + 2.*k - 3.*k.^2.*exp(-k) + 2.*k.^3.*sin(k).*exp(-k./5);
    % Evaluación de la función en el punto l=k+1.
    gl = 1 + 2.*l - 3.*l.^2.*exp(-l) + 2.*l.^3.*sin(l).*exp(-l./5);
    % Condicional que marca el subintervalo donde se encuentra un cruce por cero.
    if gk*gl < 1e-3
         % Método de punto fijo.
         [Cero,Mat] = PuntoFijo(k);
         r = r + 1;                  % Contador de cruces por cero.
         Cruce(r) = Cero;            % Almacena los cruces por cero.
         dr = length(Mat(:,1));      % Número de iteraciones para encontrar el cruce
                                     % por cero.
        dc = length(Mat(1,:));       % Número de variables por almacenar.
```

www.FreeLibros.me

```
        M(r,1:dr,1:dc) = Mat;       % Matriz que almacena todas las iteraciones de
                                    % todos los cruces por cero.
    end
end
M1(:,:) = M(1,:,:);    % Iteraciones del primer cruce por cero.
M2(:,:) = M(2,:,:);    % Iteraciones del segundo cruce por cero.
M3(:,:) = M(3,:,:);    % Iteraciones del tercer cruce por cero.
M4(:,:) = M(4,:,:);    % Iteraciones del cuarto cruce por cero.
M5(:,:) = M(5,:,:);    % Iteraciones del quinto cruce por cero.
```

Función de Matlab llamada punto fijo

```
%  Función del método Punto fijo para cálculo de cruces por cero de una función no
%  lineal que se mueve en plano real.
%
%  El método calcula un cruce por cero.
%
%  La función se llama de la siguiente manera:
%
%               [Cero,Mat]=Punto fijo(a).
%
%  Entradas:
%               a  -- Punto inicial del método.
%
%  Salida:
%         Cero -- Valor de la variable para la cual la magnitud de la función es cero
%         o menor a una tolerancia especificada previamente.
%         Mat  -- Vector que contiene todas las iteraciones hasta convergencia.
%
function[Cero,Mat] = PuntoFijo(a);
Err = 1;             % Inicializa el error para ingresar al ciclo iterativo.
tol = 1e-6;          % Tolerancia especificada para la convergencia.
d = round(a/pi);     % Numero de pi enteros de acuerdo a la condición inicial.
e = (-1)^d;          % Signo de acuerdo si es par o impar en # de pi.
% Valor de la función en el punto inicial.
fa = 1 + 2.*a - 3.*a.^2.*exp(-a) + 2.*a.^3.*sin(a).*exp(-a./5);
% Contador de iteraciones.
c   = 1;
% Inicia la matriz que almacena cada iteración.
Mat(c,:) = [a fa];
while Err > tol & c < 20
   % Cálculo del punto de aproximación que determina el método de punto fijo.
   gx = d*pi+e*asin((-1 -2.*a +3.*a.^2.*exp(-a))/(2.*a.^3.*exp(-a./5)));
   % Valor de la función en el nuevo punto.
   fgx = 1 + 2.*gx - 3.*gx.^2.*exp(-gx) + 2.*gx.^3.*sin(gx).*exp(-gx./5);
   % Contador de iteraciones para no dejar en un ciclo el programa en caso de alguna
   % inconsistencia.
   c = c + 1;
   % Matriz que almacena los resultados de cada iteración.
   Mat(c,:) = [gx fgx];
   % El nuevo punto para la siguiente iteración es
   a  = gx;
   % Criterio de error.
   Err = abs(fgx);
end
% Cruce por cero que determina el método de la secante.
Cero = gx;
```

2.10.5 Método de Newton-Raphson

El método de Newton-Raphson se basa en una aproximación por incrementos equivalente a una expansión en series de Taylor hasta la primera derivada. En Matlab, cuando se usa el signo % significa que se está haciendo un comentario. Adicionalmente las funciones propias de Matlab aparecen sombreadas.

www.FreeLibros.me

Programa principal del método de Newton-Raphson

```
%  Programa principal para determinar los cruces por cero de una función continua.
clear all
clc
%  Utilizando el método de Newton-Raphson, determinar los cruces por cero de la
%  función.
%  fx = 1 + 2.*x - 3.*x.^2.*exp(-x) + 2.*x.^3.*sin(x).*exp(-x./5) dentro del
%  intervalo [4,20]. Utilizar un error relativo de 1e-6.
r=0;       % Inicia el contador de cruces por cero.
for k = 4:20
    l = k+1;
    % Evaluación de la función en el punto k.
    gk = 1 + 2.*k - 3.*k.^2.*exp(-k) + 2.*k.^3.*sin(k).*exp(-k./5);
    % Evaluación de la función en el punto l=k+1.
    gl = 1 + 2.*l - 3.*l.^2.*exp(-l) + 2.*l.^3.*sin(l).*exp(-l./5);
    % Condicional que marca el subintervalo donde se encuentra un cruce por cero.
    if gk*gl < 1e-3
         % Metodo de Newton-Raphson.
         [Cero,Mat] = Newton-Raphson(k);
         r = r + 1;                    % Contador de cruces por cero.
         Cruce(r) = Cero;              % Almacena los cruces por cero.
        dr = length(Mat(:,1));         % Número de iteraciones para encontrar el cruce
                                       % por cero.
        dc = length(Mat(1,:));         % Número de variables por almacenar.
        M(r,1:dr,1:dc) = Mat;          % Matriz que almacena todas las iteraciones de
                                       % todos los cruces por cero.
    end
end
M1(:,:) = M(1,:,:);    % Iteraciones del primer cruce por cero.
M2(:,:) = M(2,:,:);    % Iteraciones del segundo cruce por cero.
M3(:,:) = M(3,:,:);    % Iteraciones del tercer cruce por cero.
M4(:,:) = M(4,:,:);    % Iteraciones del cuarto cruce por cero.
M5(:,:) = M(5,:,:);    % Iteraciones del quinto cruce por cero.
```

Función de Matlab llamada Newton-Raphson

```
%  Función del método de Newton-Raphson para cálculo de cruces por cero de una
%  función no lineal que se mueve en plano real.
%
%  El método calcula un cruce por cero.
%
%  La función se llama de la siguiente manera.
%
%          [Cero,Mat]=Newton-Raphson(a)
%
%  Entradas:
%          a  -- Punto inicial del método.
%
%  Salida:
%          Cero -- Valor de la variable para la cual la magnitud de la función es cero
%          o menor a una tolerancia especificada previamente.
%          Mat  -- Vector que contiene todas las iteraciones hasta convergencia.
%
function[Cero,Mat] = NewtonRaphson(a);
Err = 1;          % Inicializa el error para ingresar al ciclo iterativo.
tol = 1e-12;      % Tolerancia especificada para la convergencia.
c   = 0;          % Inicializa el contador de iteraciones.
while Err > tol & c < 20
   % Valor de la función en el punto inicial.
   fa  = 1 + 2.*a - 3.*a.^2.*exp(-a) + 2.*a.^3.*sin(a).*exp(-a./5);
   % Valor de la derivada de la función en el punto inicial.
   fpa = 2 - 6.*a.*exp(-a) + 3.*a.^2.*exp(-a) + 6.*a.^2.*sin(a).*exp(-a./5) + ...
         2.*a.^3.*cos(a).*exp(-a./5) - (2/5).*a.^3.*sin(a).*exp(-a./5);
   % Cálculo del nuevo valor de x dado por el método de Newton-Raphson.
```

www.FreeLibros.me

```
   xn = a - fa/fpa;
   % Valor de la funcion en el nuevo punto.
   fxn = 1 + 2.*xn - 3.*xn.^2.*exp(-xn) + 2.*xn.^3.*sin(xn).*exp(-xn./5);
   % Contador de iteraciones para no dejar en un ciclo el programa en caso de alguna
   % inconsistencia.
   c = c + 1;
   % Matriz que almacena los resultados de cada iteración.
   Mat(c,:) = [a fa fpa];
   % Asignación de la aproximación más nueva para seguir la iteración.
   a  = xn;
   % Criterio de error.
   Err = abs(fxn);
end
%  Cruce por cero que determina el método de Newton-Raphson.
Cero = xn;
```

2.10.6 Método de Newton-Raphson para sistemas de ecuaciones

El método de Newton-Raphson para sistemas de ecuaciones no lineales es una extensión del método de Newton-Raphson para una ecuación no lineal, se basa en el mismo principio de incrementos y se puede obtener como una expansión en series de Taylor de primer orden para sistemas matriciales. En Matlab, cuando se usa el signo % significa que se está haciendo un comentario; adicionalmente, las funciones propias de Matlab aparecen sombreadas.

Programa principal del método de Newton-Raphson para sistema de ecuaciones

```
%  Programa principal para resolver un sistema de ecuaciones no lineales utilizando
%  el método de Newton-Raphson.
%  Cada sistema de ecuaciones se resuelve en forma única. El que se resuelve aquí es
%  2x - 3xy + 2z^2 = 1
%   x + 7y  + 2yz  = 2
%  3x +  xy + 8z   = 3
clear all
clc
x(1)=1;   y(1)=1;   z(1)=1;   % Se fijan las condiciones iniciales.
Err = 1;                      % Inicializa el error para ingresar al ciclo iterativo.
tol = 1e-6;                   % Tolerancia especificada para la convergencia.
k   = 1;                      % Inicializa el contador de iteraciones.
while Err > tol & k < 20
    % Contador de iteraciones para no dejar en un ciclo el programa en caso de alguna
    % inconsistencia.
    k = k+1;
    % Evaluación del jacobiano.
    J = [2-3*y(k-1)     -3*x(k-1)    4*z(k-1)
                 1     7+2*z(k-1)    2*y(k-1)
          3+y(k-1)        x(k-1)    8       ];
    % Evaluación del sistema de ecuaciones no lineales.
    F0 = [2*x(k-1)    - 3*x(k-1)*y(k-1)    + 2*z(k-1)*z(k-1)    - 1
            x(k-1)           + 7*y(k-1)    + 2*y(k-1)*z(k-1)    - 2
          3*x(k-1)      + x(k-1)*y(k-1)           + 8*z(k-1)    - 3 ];
    % Incrementos de cada variable calculados por Newton-Raphson.
    Dx = -inv(J)*F0;
    % Cálculo de los nuevos valores de las variables dadas por el método.
    x(k) = Dx(1) + x(k-1);
    y(k) = Dx(2) + y(k-1);
    z(k) = Dx(3) + z(k-1);
    % Matriz que almacena los resultados de todas las iteraciones.
    MD = [x; y; z;];
    % Cálculo del error para cumplir la tolerancia.
    Err = max(abs(MD(:,k)-MD(:,k-1)));
end
```

www.FreeLibros.me

2.10.7 Método de punto fijo multivariable; Gauss y Gauss-Seidel

El método de punto fijo multivariable es una extensión del método de punto fijo, sólo aplicado a un grupo de ecuaciones no lineales. El método tiene dos variantes, la de Gauss que sustituye todos los valores en todas las ecuaciones en forma simultánea y el de Gauss-Seidel que va utilizando los valores más nuevos en cada cálculo. En Matlab, cuando se usa el signo % significa que se está haciendo un comentario. Adicionalmente las funciones propias de Matlab aparecen sombreadas.

Programa principal del método de punto fijo multivariable

```
%  Programa principal para resolver un sistema de ecuaciones no lineales utilizando
%  el método de Punto fijo multivariable, ya sea por el método de Gauss o por el
%  método de Gauss-Seidel.
%
%  Cada sistema de ecuaciones se resuelve en forma única. El que se resuelve aquí es
%
%  2x - 3xy + 2z^2 = 1
%   x + 7y  + 2yz  = 2
%  3x +  xy + 8z   = 3
%  Solución del sistema de ecuaciones no lineales mediante el método de Punto fijo
%  multivariable utilizando la sustitución dada por el método de Gauss.
clear all
clc
x(1)=1;   y(1)=1;   z(1)=1;    % Se fijan las condiciones iniciales.
Err = 1;                       % Inicializa el error para ingresar al ciclo iterativo.
tol = 1e-6;                    % Tolerancia especificada para la convergencia.
k   = 1;                       % Inicializa el contador de iteraciones.
while Err > tol & k < 20
    % Contador de iteraciones para no dejar en un ciclo el programa en caso de alguna
    % inconsistencia.
    k = k+1;
    % Cálculo de los nuevos valores de las variables dadas por el método de Gauss.
    x(k) = (1 + 3*y(k-1)*x(k-1) - 2*z(k-1)^2) / 2;
    y(k) = (2 - x(k-1) - 2*z(k-1)*y(k-1)) / 7;
    z(k) = (3 - 3*x(k-1) - y(k-1)*x(k-1)) / 8;
    % Cálculo del error para cumplir la tolerancia.
    Err = max(abs([x(k)-x(k-1)   y(k)-y(k-1)   z(k)-z(k-1)]));
end
% Matriz que almacena los resultados de todas las iteraciones.
MDGauss = [x; y; z;];
%  Solución del sistema de ecuaciones no lineales mediante el método de Punto fijo
%  multivariable utilizando la sustitución dada por el método de Gauss-Seidel.
clear x y z
clc
x(1)=1;   y(1)=1;   z(1)=1;    % Se fijan las condiciones iniciales.
Err = 1;                       % Inicializa el error para ingresar al ciclo iterativo.
tol = 1e-6;                    % Tolerancia especificada para la convergencia.
k   = 1;                       % Inicializa el contador de iteraciones.
while Err > tol & k < 20
    % Contador de iteraciones para no dejar un ciclo en el programa en caso de alguna
    % inconsistencia.
    k = k+1;
    % Cálculo de los nuevos valores de las variables dadas por el método de Gauss-
    % Seidel.
    x(k) = (1 + 3*y(k-1)*x(k-1) - 2*z(k-1)^2) / 2;
    y(k) = (2 - x(k) - 2*z(k-1)*y(k-1)) / 7;
    z(k) = (3 - 3*x(k) - y(k)*x(k)) / 8;
    % Cálculo del error para cumplir la tolerancia.
    Err = max(abs([x(:,k)-x(:,k-1)   y(:,k)-y(:,k-1)   z(:,k)-z(:,k-1)]));
end
%  Matriz que almacena los resultados de todas las iteraciones.
MDGaussSeidel = [x; y; z;];
```

www.FreeLibros.me

Problemas propuestos

2.11.1 Encuentre la tercera iteración por el método de bisección de la función $f(x) = \text{sen}(x) - \cos(1 + x^2) - 1$ en el intervalo $[a, b]$, donde $a = 2\pi/3$ y $b = \pi$.

2.11.2 Encuentre la cuarta iteración por el método de bisección de la función $f(x) = x^2 \ln x - 9x - 18$ en el intervalo $[a, b]$, donde $a = 6$ y $b = 7$.

2.11.3 Encuentre la quinta iteración por el método de bisección de la función $f(x) = x^3 - 2x^2 \text{sen}(x)$ en el intervalo $[a, b]$, donde $a = 2$ y $b = 3$.

2.11.4 Encuentre la sexta iteración por el método de bisección de la función $f(x) = 2x \tan(x) - 10$ en el intervalo $[a, b]$, donde $a = 1.3$ y $b = 1.4$.

2.11.5 Encuentre la séptima iteración por el método de bisección de la función $f(x) = x \log x - 10$ en el intervalo $[a, b]$, donde $a = 6$ y $b = 5$.

2.11.6 Encuentre los dos cruces por cero por el método de bisección de la función $f(x) = 2x^{0.2} - e^{-\pi x} \tan(x) - 2$, en el intervalo $[a, b]$, donde $a = 1$ y $b = 1.55$. El proceso se detiene cuando existen dos iteraciones consecutivas que difieren en menos de 0.001 como valor absoluto. La búsqueda del intervalo que contiene un cruce por cero se hace con un $\Delta x = 0.05$.

2.11.7 Encuentre los cinco cruces por cero utilizando el método de bisección de la función $f(x) = 2x^{0.6} - \cos(x)\log_{10}(x) - 20$, en el intervalo $[a, b]$, donde $a = 40$ y $b = 55$. Detener el proceso cuando la evaluación de $f(x)$ sea menor a 0.0001 como valor absoluto. La búsqueda del intervalo que contiene un cruce por cero se hace con un $\Delta x = 0.1$.

2.11.8 Aplicando el método de regla falsa, encuentre el valor del cruce por cero de la función $f(x) = x^3 e^{-x} + 4x^2 - 10$ iniciando con $x_0 = 1$ y $x_1 = 2$. Detener el proceso en la cuarta iteración.

2.11.9 Aplicando el método de regla falsa determine el cruce por cero de la función $f(x) = x^3 \cos(x) - 5x^2 - 1$; en forma inicial el intervalo es $[36, 37]$. Detener el proceso en cuatro iteraciones.

2.11.10 Por el método de regla falsa encuentre la tercera iteración de la función $f(x) = 2x - 20\,\text{sen}(x)$ iniciando con $x_0 = 2$ y $x_1 = 4$.

2.11.11 Por el método de regla falsa encuentre la quinta iteración de la función $f(x) = 1\,000x^2 \tan(x) e^{-5x} - 10$ iniciando con $x_0 = 0.5$ y $x_1 = 1$.

2.11.12 Utilizando el método de regla falsa encuentre los tres cruces por cero de la función $f(x) = x^3 - 2x^2 \cos(2x) - 12$ dentro del intervalo $[a, b]$, donde $a = 0$ y $b = 4$. Detener el proceso cuando la evaluación de $f(x)$ sea menor que 0.0001 como valor absoluto.

2.11.13 Utilizando el método de regla falsa encuentre los tres cruces por cero de la función $f(x) = 2\,000x^3 e^{-5x} \cos(2x) + 1$, dentro del intervalo $[a, b]$, donde $a = -0.1$ y $b = 3$. Detener el proceso cuando la evaluación de $f(x)$ sea menor que 0.0001 como valor absoluto.

2.11.14 Aplique el método de la secante iniciando con $x_0 = 1$, $x_1 = 2$, $f(x_0) = 2$ y $f(x_1) = 1.5$; con estos valores cuál es el valor de x_2.

2.11.15 Realice cuatro iteraciones utilizando el método de la secante a la función $f(x) = x - 2\cos(x)$ con $x_0 = 1$ y $x_1 = 1.5$ como condiciones iniciales.

2.11.16 Realice el número de iteraciones hasta la convergencia, utilizando el método de la secante a la función $f(x) = 5xe^{-x} + \cos(5x)$ con $x_0 = 3.9$ y $x_1 = 4$ como condiciones iniciales. Utilizar como criterio de convergencia cuando la función evaluada $f(x)$ sea menor que 0.0001, en valor absoluto.

2.11.17 Realice el número de iteraciones hasta la convergencia, utilizando el método de la secante a la función $f(x) = x^2 \cos(10x) + 1$ con $x_0 = 1.3$ y $x_1 = 1.4$ como condiciones iniciales. Utilizar como criterio de convergencia cuando la función evaluada $f(x)$ sea menor que 0.0001, en valor absoluto.

www.FreeLibros.me

2.11.18 Realice el número de iteraciones hasta la convergencia, utilizando el método de la secante a la función $f(x)=23\,000x^3e^{-0.01x}+\text{sen}(5\,000x)$ con $x_0=0.0144$ y $x_1=0.0145$ como condiciones iniciales. Utilizar como criterio de convergencia cuando la función evaluada $f(x)$ sea menor que 0.0001, en valor absoluto.

2.11.19 Realice el número de iteraciones hasta la convergencia, utilizando el método de la secante a la función $f(x)=5x\cos(x)+2$ con $x_0=1.0$ y $x_1=1.1$ como condiciones iniciales. Utilizar como criterio de convergencia cuando la función evaluada $f(x)$ sea menor que 0.0001, en valor absoluto.

2.11.20 Realice el número de iteraciones hasta la convergencia, utilizando el método de la secante a la función $f(x)=e^{-2x}\,\text{sen}(0.1x)-e^{-10x}\cos(0.1x)+0.3$ con $x_0=0.4$ y $x_1=0.3$ como condiciones iniciales. Utilizar como criterio de convergencia cuando la función evaluada $f(x)$ sea menor que 0.0001, en valor absoluto.

2.11.21 Por el método de punto fijo encuentre la quinta iteración de la función $f(x)=8x+\cos(x+\pi)-x^2$ con $x_0=2$ como condición inicial.

2.11.22 Con el método de punto fijo, utilizando el arreglo $g(x)=\ln x-\frac{x^2}{3}+\frac{5}{3}$ con $x_0=1.2$, encuentre x_4.

2.11.23 Con el método de punto fijo determine el cruce por cero de la función $f(x)=x^2e^{-x}+\ln x-3$ iniciando con $x_0=1$ y con $x_0=50$ como condición inicial.

2.11.24 Con el método de punto fijo determine el cruce por cero de la función $f(x)=\text{sen}(0.1x)+e^{-2x}-0.6$, en el intervalo $[20, 30]$.

2.11.25 Con el método de punto fijo determine el cruce por cero de la función $f(x)=\cos(3x)+5e^{-0.01x}-3$, en el intervalo $[34, 34.5]$.

2.11.26 Por el método de Newton-Raphson determine el cruce por cero de la función $f(x)=x^2\cos x-6x\ln x-25$ con $x_0=23$ como condición inicial.

2.11.27 Aplicando el método de Newton-Raphson determine el cruce por cero de la función $f(x)=(x-2)^2-\ln x$ con $x_0=1$ como condición inicial.

2.11.28 Aplique el método de Newton-Raphson para determinar el cruce por cero de la función $f(x)=\cos(x)-3x$ con $x_0=0.5$ como condición inicial; obtener la convergencia con cuatro cifras decimales.

2.11.29 Aplique el método de Newton-Raphson para determinar el cruce por cero de la función $f(x)=\cos(5x)+5\,\text{sen}(15x)+e^{-0.05x}-3$ contenido en el intervalo $[16.1, 16.5]$; obtener la convergencia con cuatro cifras decimales. Encuentre la condición inicial apropiada.

2.11.30 Aplique el método de Newton-Raphson para determinar el cruce por cero de la función $f(x)=\log_{10}(50x-40)-x^{1.2}+20\cos(x)+12$ contenido en el intervalo $[5, 10]$; obtener convergencia con cuatro cifras decimales. Encuentre la condición inicial apropiada.

2.11.31 Aplique el método de Newton-Raphson para resolver un sistema de ecuaciones no lineales tomando como condiciones iniciales $x_0=1$, $y_0=1$ y $z_0=1$. Detener el proceso hasta que el valor de todos los incrementos sea menor que $\Delta<1e^{-6}$, el sistema de ecuaciones es el siguiente:

$$
\begin{aligned}
71xz-9x^3+7z^4&=240\\
xy+17yz+xyz&=31\\
3xz+3yz+13z^2&=132
\end{aligned}
$$

2.11.32 Aplique el método de Newton-Raphson para resolver un sistema de ecuaciones no lineales tomando como condiciones iniciales $x_0=10$, $y_0=10$ y $z_0=10$. Detenga el proceso cuando el valor de todos los incrementos sea menor que $\Delta<1e^{-6}$. El sistema de ecuaciones es el siguiente:

www.FreeLibros.me

$$24xz+3y^2-5xyz^2=34$$
$$x-57y^2-12xz=53$$
$$4xy+3yz^3+25xz^2=39$$

2.11.33 Aplique el método de Newton-Raphson para resolver el sistema de ecuaciones no lineales siguiente:

$$xyz-3x^2y^2+2yz^2=45$$
$$xz^2+7y^2-5yz=76$$
$$4xy^3+2yz^2+3xz^3=91$$

Realice el proceso tomando como condiciones iniciales $x_0=1$, $y_0=1$ y $z_0=1$. Repítalo, pero tomando ahora como condiciones iniciales $x_0=1$, $y_0=2$ y $z_0=3$. Detenga el proceso cuando el valor de todos los incrementos sea menor que $\Delta<1e^{-6}$.

2.11.34 Analizando los resultados del problema anterior (2.11.33), explique la diferencia entre los resultados dependiendo de la condición inicial que se utilice.

2.11.35 Aplique el método de punto fijo multivariable en sus dos variantes (Gauss y Gauss-Seidel) para resolver el sistema de ecuaciones no lineales siguiente:

$$7x+xy+4z=24$$
$$x^2+28y-3yz=12$$
$$-3xy+yz+13z=43$$

Realice el proceso tomando como condiciones iniciales $x_0=4$, $y_0=7$ y $z_0=3$. Repítalo, pero tomando ahora como condiciones iniciales $x_0=4$, $y_0=7$ y $z_0=8$. Detenga el proceso cuando el valor de todos los incrementos sea menor que $\Delta<1e^{-6}$. Interprete y explique los resultados de cada condición inicial.

www.FreeLibros.me

Capítulo 3

Solución de ecuaciones polinomiales

3.1 Introducción

El problema que se considerará en este capíulo es determinar las raíces de un polinomio que tiene la siguiente estructura:

$$f(z) = a_0 + a_1 z + \cdots + a_n z^n \quad \text{con} \quad a_n \neq 0 \tag{3.1a}$$

o bien,

$$f(z) \equiv a_n (z - z_1)(z - z_2)\cdots(z - z_n) \tag{3.1b}$$

donde los a_j, $j = 0, 1, \ldots, n$ son coeficientes reales y los z_r, $r = 1, 2, \ldots, n$ son, en general, raíces complejas. Al número α se le llama raíz de la ecuación si $f(\alpha) = 0$. El método más sencillo es aquel que encuentra solamente las raíces reales; sólo que para tener una solución completa se deben encontrar todas las raíces. Ya se ha demostrado que existen exactamente n raíces para un polinomio de grado n. Se debe tener siempre en mente que las *raíces complejas ocurren en pares complejos conjugados*. Así, para cualquier raíz $a + ib$ existe la raíz conjugada correspondiente $a - ib$.

A simple vista parece no haber razones para suponer que habría una dificultad particular para encontrar las raíces de un polinomio. *Analíticamente, un polinomio es una función muy simple, ya que todas sus derivadas son continuas en cualquier región y se pueden integrar fácilmente*. Sin embargo, el problema de calcular aproximaciones precisas de todas las raíces de un polinomio puede ser extremadamente difícil, incluso para polinomios de grado tan bajo como 20. El problema se complica cuando se presentan raíces repetidas.

Un ejemplo sencillo ilustra cómo una pequeña variación en los coeficientes de un polinomio puede causar grandes variaciones en las raíces. El siguiente polinomio con raíces $z_k = k$ parece simple a primera vista,

$$f(z) = (z - 1)(z - 2)\cdots(z - 20) \tag{3.2}$$

Desarrollando la ecuación se obtiene,

$$f(z) = z^{20} - 210z^{19} + 20\,615z^{18} + \cdots + 20! \tag{3.3}$$

www.FreeLibros.me

Sin embargo, un cambio tan pequeño como agregar (-2^{-23}) al coeficiente −210, que representa un pequeño cambio en la séptima cifra decimal significativa, que queda con un valor de −210.000000119209, cambia las raíces $z_{10},\ldots,z_{19}$ a raíces complejas. También hay cambios significativos en todos los valores. Las nuevas raíces se proporcionan en la tabla 3.1 y son:

Tabla 3.1 Raíces nuevas después del pequeño cambio en el coeficiente.

Raíces nuevas	$z_1 = 1.00000000000013$	$z_{10} = 10.095216877102 - 0.642140573079565i$
	$z_2 = 2.00000000000384$	$z_{11} = 10.095216877102 + 0.642140573079565i$
	$z_3 = 2.99999999923892$	$z_{12} = 11.7933291579272 - 1.65214479613009i$
	$z_4 = 4.00000001968083$	$z_{13} = 11.7933291579272 + 1.65214479613009i$
	$z_5 = 4.99999978833186$	$z_{14} = 13.9922919681938 - 2.51887761175445i$
	$z_6 = 6.00000577000426$	$z_{15} = 13.9922919681938 + 2.51887761175445i$
	$z_7 = 6.99972684950338$	$z_{16} = 16.730744522784 - 2.81265091036364i$
	$z_8 = 8.00699644364665$	$z_{17} = 16.730744522784 + 2.81265091036364i$
	$z_9 = 8.91829273399838$	$z_{18} = 19.5024493611409 - 1.94033764150186i$
	$z_{20} = 20.8469147405052$	$z_{19} = 19.5024493611409 + 1.94033764150186i$

3.2 Aritmética para polinomios

Una ventaja de trabajar con polinomios, es que existen algoritmos simples para ejecutar muchos de los cálculos necesarios. En esta sección se aborda el problema de encontrar, en cualquier punto dado α, el valor de un polinomio, su derivada, y el residuo de dividirlo entre $z-\alpha$. Los procedimientos que se describen son la multiplicación anidada, la división sintética y la evaluación de la derivada.

3.2.1 Multiplicación anidada

En la aritmética computacional, se señala que la suma y la multiplicación son la base de las operaciones por computadora, pero se emplean otras funciones por segmentos de código que son muy lentos. Esto es cierto para funciones del tipo exponencial tales como z^r; por tanto, sería conveniente evitar tales funciones. La evaluación directa de un polinomio requiere de $n-1$ potenciaciones, n multiplicaciones y n sumas.

El *método de multiplicación anidada* aplicado a una función cúbica tiene la siguiente forma:

$$((a_3 z + a_2)z + a_1)z + a_0 \tag{3.4}$$

Se puede notar que el efecto de esto es la multiplicación de a_3 por z^3, etc. Esto se puede escribir en forma algorítmica como sigue:

$$\begin{aligned} b_n &= a_n \\ b_r &= z b_{r+1} + a_r \qquad r = n-1, n-2, \ldots, 1, 0 \end{aligned} \tag{3.5a,b}$$

lo cual es fácil de programar en una computadora. La cantidad b_0 da el valor del polinomio para un valor dado de z [Mathews *et al.*, 2000].

EJEMPLO 3.1

Como ejemplo se considera el valor del polinomio cúbico $2z^3 - z^2 + 6$ evaluado en el punto $z = 1.1$. La formulación dada por la ecuación (3.5a) indica que el primer término es $b_3 = 2.00$.

SOLUCIÓN. La ecuación (3.5b) es una función recursiva; si se aplica tomando el resultado de b_3 como el valor de la b_{r+1} se obtiene,

$$b_2 = z b_3 + a_2 = (1.10)(2.00) - 1.00 = 1.20$$

www.FreeLibros.me

Aplicando nuevamente (3.5b) se llega a:

$$b_1 = zb_2 + a_1 = (1.10)(1.20) + 0.00 = 1.32$$

Por último, se obtiene el valor buscado como:

$$b_0 = zb_1 + a_0 = (1.10)(1.32) + 6.00 = 7.4520$$

El resultado de evaluar el polinomio en la forma tradicional en $z = 1.1$ es:

$$r = 2(1.1)^3 - (1.1)^2 + 6 = 7.4520$$

Este resultado comprueba que la multiplicación anidada no es un método que introduzca errores en el resultado y que, por supuesto, evita la evaluación de potencias.

3.2.2. División sintética

El proceso de división de un polinomio por un factor $z-\alpha$ es importante por dos razones. Esto forma un componente del esquema para encontrar las raíces de un polinomio y también habilita el teorema de residuo del álgebra para emplearlo eficientemente cuando se calcula utilizando computadora [Burden *et al.* 1985]. El teorema del residuo se desarrolla escribiendo un polinomio de la forma:

$$f_n(z) = (z-\alpha) f_{n-1}(z) + R \tag{3.6}$$

El subíndice denota el grado del polinomio y la división dará un cociente de grado $n-1$ y un residuo R que es una constante. Si se emplea $z=\alpha$, entonces $R = f_n(\alpha)$, de manera que el residuo dará el valor del polinomio en $z=\alpha$. Primero se muestra una división muy larga entre $z-\alpha$ y después la tabla abreviada que se conoce como *división sintética*.

$$\begin{array}{r l}
 & \left(a_3z^2 + (a_2 + a_3\alpha)z + a_1 + (a_2 + a_3\alpha)\alpha\right) \\
z-\alpha \big) & a_3z^3 + a_2z^2 + a_1z + a_0 \\
 & \underline{a_3z^3 - a_3\alpha z^2} \\
 & \quad (a_2 + a_3\alpha)z^2 + a_1z \\
 & \quad \underline{(a_2 + a_3\alpha)z^2 - (a_2 + a_3\alpha)\alpha z} \\
 & \qquad [a_1 + (a_2 + a_3\alpha)\alpha]z + a_0 \\
 & \qquad \underline{[a_1 + (a_2 + a_3\alpha)\alpha]z - [a_1 + (a_2 + a_3\alpha)\alpha]\alpha} \\
 & \qquad\qquad a_0 + [a_1 + (a_2 + a_3\alpha)\alpha]\alpha
\end{array} \tag{3.7}$$

Se puede observar que escribir las potencias de z en (3.7) es superfluo; basta colocar los coeficientes en la columna correcta, y así la parte izquierda de la columna, que produce ceros en la resta, se ignora. Se puede lograr una simplificación adicional cambiando el signo de α y sumando las dos cantidades. Así se produce la siguiente tabla:

$$\begin{array}{c|c l l l l}
 & & a_3 & a_2 & a_1 & a_0 \\
+\alpha & + & 0 & p_3\alpha & p_2\alpha & p_1\alpha \\
\hline
 & & p_3 = a_3 & p_2 = a_2 + p_3\alpha & p_1 = a_1 + p_2\alpha & p_0 = a_0 + p_1\alpha
\end{array} \tag{3.8}$$

El elemento inferior de la última columna del esquema da el residuo (p_0), por ejemplo, el valor del polinomio en $z=\alpha$. Los coeficientes del cociente del polinomio están dados por p_3, p_2 y p_1. La sección 3.7.1 proporciona el programa en Matlab para realizar la división sintética con un factor simple.

www.FreeLibros.me

EJEMPLO 3.2

Con el esquema dado por (3.8), sacar las raíces de la siguiente ecuación polinomial,

$$f(z) = z^5 - z^4 - 60z^3 - 20z^2 + 464z - 384$$

Solución. Utilizando el esquema de la ecuación (3.8) y tomando como primera raíz $\alpha = 8$ se llega a:

$$\begin{array}{r|rrrrrr} & 1 & -1 & -60 & -20 & 464 & -384 \\ +8 & +\;0 & 8 & 56 & -32 & -416 & 384 \\ \hline & p_5 = 1 & p_4 = 7 & p_3 = -4 & p_2 = -52 & p_1 = 48 & p_0 = 0 \end{array}$$

Así, la primera raíz es $z_1 = 8$, el residuo es $p_0 = 0$ y el cociente es:

$$C_1(z) = z^4 + 7z^3 - 4z^2 - 52z + 48$$

Para que con el método anterior se obtenga una raíz del polinomio, es necesario que con el proceso se llegue siempre a un valor $p_0 = 0$. De otra forma la solución propuesta no es raíz del polinomio.

3.2.2.1 *División sintética de un factor cuadrático*

El procedimiento anterior tiene los mismos pasos computacionales que la multiplicación anidada, pero este esquema también se puede extender a la división utilizando un factor cuadrático, que es el proceso usado en la solución del polinomio cuando se tiene un par complejo conjugado de raíces. La división sintética mediante un factor cuadrático $z^2 + \alpha z + \beta$ tiene la forma siguiente:

$$\begin{array}{r|lllll} -\alpha & a_4 & a_3 & a_2 & a_1 & a_0 \\ & & -p_4\alpha & -p_3\alpha & -p_2\alpha & \\ -\beta & & & -p_4\beta & -p_3\beta & -p_2\beta \\ \hline & p_4 = a_4 & p_3 = a_3 - p_4\alpha & p_2 = a_2 - p_3\alpha - p_4\beta & p_1 = a_1 - p_2\alpha - p_3\beta & p_0 = a_0 - p_2\beta \end{array} \quad \textbf{(3.9)}$$

En este caso, el residuo es $R = p_1 z + p_0$ y los coeficientes del cociente son p_4, p_3 y p_2. La sección 3.7.2 proporciona el programa Matlab para realizar la división sintética utilizando un factor cuadrático.

EJEMPLO 3.3

Determinar el cociente del polinomio $f(z) = z^4 + 7z^3 - 15z^2 - 121z - 520$, si se sabe que tiene un factor cuadrático $f(z) = z^2 + 4z + 13$ que contiene dos raíces complejas conjugadas.

Solución. El esquema dado por la ecuación (3.9) lleva al resultado siguiente,

$$\begin{array}{r|lrrrr} -4 & 1 & 7 & -15 & -121 & -520 \\ & & -4 & -12 & 160 & 0 \\ -13 & & & -13 & -39 & \\ \hline & p_4 = 1 & p_3 = 3 & p_2 = -40 & p_1 = 0 & p_0 = 0 \end{array}$$

Si el residuo del factor cuadrático está dado por $R = p_1 z + p_0$, se obtiene cero como residuo, y como cociente se obtiene la ecuación $C_1(z) = z^2 + 3z - 40$.

www.FreeLibros.me

3.2.3 Evaluación de la derivada

Consideremos la fórmula (3.6) que muestra la división entre $z-\alpha$. Por ejemplo:

$$f_n(z)=(z-\alpha)f_{n-1}(z)+R \quad \textbf{(3.10)}$$

derivando ambos términos, recordando que R es una constante, se tiene:

$$f_n'(z)=f_{n-1}(z)+(z-\alpha)f_{n-1}'(z) \quad \textbf{(3.11)}$$

Sustituyendo el valor de $z=\alpha$, el segundo término del lado derecho de la ecuación es cero y la derivada se obtiene por el valor de $f_{n-1}(z)$. Este polinomio esta disponible en el proceso de la división sintética y se puede evaluar en el punto $z=\alpha$ empleando una división sintética adicional o por la multiplicación anidada.

El proceso descrito utiliza el método de Newton para un polinomio sencillo, dado que el proceso se puede usar para evaluar $f(x)$ y $f'(x)$.

EJEMPLO 3.4

Utilizando el procedimiento descrito en esta sección, determinar la derivada en $z=5$ del polinomio $f(z)=z^5-z^4-43z^3+61z^2+342z-360$.

Solución. Primero se aplica la división sintética para obtener el cociente, así se obtiene:

	1	−1	−43	61	342	−360
+5	0	5	20	−115	−270	360
	$p_5=1$	$p_4=4$	$p_3=-23$	$p_2=-54$	$p_1=72$	$p_0=0$

Por tanto, $f_{n-1}(z)=z^4+4z^3-23z^2-54z+72$, con residuo igual a cero. Evaluando $f_{n-1}(z)$ mediante el proceso de multiplicación anidada se obtiene la secuencia:

$$b_4=a_4 \qquad =1$$

$$b_3=zb_4+a_3=(5)(1)+4=9$$

$$b_2=zb_3+a_2=(5)(9)-23=22$$

$$b_1=zb_2+a_1=(5)(22)-54=56$$

$$b_0=zb_1+a_1=(5)(56)+72=352$$

Por tanto, la derivada de la función es,

$$f'(z)=b_0=352$$

Comprobación. Si se quiere comprobar el resultado anterior, se toma la ecuación original de $f(z)$, se saca su derivada y se evalúa en $z=5$; así, se tiene que la derivada es,

$$f'(z)=5z^4-4z^3-129z^2+122z+342$$

Evaluando en $z=5$ se obtiene,

$$f'(z)=5(5)^4-4(5)^3-129(5)^2+122(5)+342=352$$

Con esto se comprueba que el procedimiento es exacto cuando el residuo es cero.

www.FreeLibros.me

3.3 Aproximaciones iniciales

Antes de anotar los datos en la computadora es posible hacer algún trabajo preliminar en la ecuación. Luego se pueden usar métodos donde interviene la intuición y habilidad para resolver el problema. El método obvio es esbozar la gráfica del polinomio y, si es posible, estimar la posición de las raíces. Existen propiedades asociadas a los polinomios que ayudan a estimar el valor aproximado de las raíces; a continuación se presentan algunas de ellas.

3.3.1 Propiedades de los polinomios

Los polinomios se pueden presentar en cualquiera de las siguientes dos formas:

$$f(z)=a_n z^n + a_{n-1}z^{n-1}+\cdots+a_1 z + a_0 \tag{3.12}$$

o bien,

$$f(z)=a_n(z-z_1)(z-z_2)\cdots(z-z_n) \tag{3.13}$$

donde $z_r\,(r=1, 2,..., n)$ son las raíces de la ecuación. Si los $a_r\,(r=1, 2,..., n)$ son reales, entonces todas las raíces también lo son, o son pares complejos conjugados. Comparando las formulaciones de las ecuaciones (3.12) y (3.13) se obtienen relaciones entre los coeficientes a_r y los *productos simétricos de las raíces*. Los tres más importantes son:

$$\sum_{r=1}^{n} z_r = \frac{-a_{n-1}}{a_n} \tag{3.14}$$

$$\sum_{r=1}^{n}\ \sum_{s=r+1}^{n} z_r z_s = \frac{a_{n-2}}{a_n} \tag{3.15}$$

$$\prod_{r=1}^{n} z_r = (-1)^n \frac{a_0}{a_n} \tag{3.16}$$

Las dos primeras ecuaciones se pueden usar para dar el límite superior de las raíces del polinomio, si todas las raíces son reales.

Se debe notar que la máxima raíz $x_{\text{máx}}$ satisface la relación:

$$\begin{aligned} x_{\text{máx}}^2 &\le x_1^2 + x_2^2 + \cdots + x_n^2 \\ &= (x_1 + x_2 + \cdots + x_n)^2 - 2(x_1x_2 + x_1x_3 + \cdots + x_1x_n + x_2x_3 + x_2x_4 + \cdots) \\ &= \left(\frac{a_{n-1}}{a_n}\right)^2 - 2\frac{a_{n-2}}{a_n} \end{aligned} \tag{3.17}$$

Así, se tiene que,

$$|x_{\text{máx}}| \le \sqrt{(a_{n-1}/a_n)^2 - 2(a_{n-2}/a_n)} \tag{3.18}$$

Es útil poder corroborar la relación (3.18) para el caso de raíces múltiples, ya que a menudo se tiene dificultad para resolverlas. Esto se puede hacer fácilmente si la raíz también es raíz de la derivada sucesiva del polinomio. Si se acepta que α es una raíz de multiplicidad k, por ejemplo:

$$f(x)=(x-\alpha)^k g(x), \tag{3.19}$$

entonces, diferenciando $f(x)$ se nota que α también es raíz de $f'(x)$, así se tiene que,

$$f'(x)=k(x-\alpha)^{k-1} g(x) + (x-\alpha)^k g'(x) \tag{3.20}$$

www.FreeLibros.me

Diferenciaciones adicionales muestran que todas las derivadas superiores a $f^{k-1}(x)$ también tienen una raíz en $x=\alpha$ y esto indica el orden de multiplicidad de la raíz por encontrar. Existe una regla simple, conocida como la *regla de signos de Descartes*, que algunas veces proporciona información útil sobre la posición de las raíces. El método tiene la desventaja de que, algunas veces, al final no da información. Si todos los coeficientes de la ecuación (3.12) son reales, entonces el número de raíces positivas de $f(z)$ es igual al número de cambios de signo de la sucesión $a_0,\ a_1,\ldots,\ a_n$, o es menor en un número par. El número de raíces negativas se relaciona en forma similar con el polinomio insertando el valor de $-x$ en el polinomio $f(x)$. El número de raíces negativas de $f(x)$ es igual al número de cambios de signo en el polinomio $f(-x)$ o menor en un número par. Entonces se puede notar que, si no hay cambios de signo en $f(x)$, no hay raíces positivas. Sin embargo, cuando los coeficientes cambian muchas veces, se pueden descubrir muy pocos valores.

3.3.2 Sucesión Sturm

En el caso simple, una sucesión de polinomios se produce como sigue; $f_0(x)$ es el polinomio original y $f_1(x)$ es la derivada de este polinomio. Los polinomios subsecuentes se definen como el negativo del residuo que se obtiene de dividir $f_r(x)$ entre $f_{r+1}(x)$.

$$f_0(x)=f(x) \tag{3.21a}$$

$$f_1(x)=f'(x) \tag{3.21b}$$

$$f_0(x)=f_1(x)q_1(x)-f_2(x) \tag{3.21c}$$

$$f_1(x)=f_2(x)q_2(x)-f_3(x) \tag{3.21d}$$

Si no hay raíces repetidas, la sucesión termina con el polinomio $f_k(x)(k=n)$, que es constante. Cuando se presentan raíces repetidas, el proceso termina con el polinomio $f_k(x)(k<n)$ y el valor $(n-k+1)$ representa la multiplicidad de la raíz.

Suponiendo que no hay raíces repetidas; se escogen dos valores a y b con $(a<b)$ y los valores de las funciones de la sucesión Sturm se evalúan en ambos puntos. Si $N(a)$ representa el número de cambios de signo en la sucesión $f_0(b),\ f_1(b),\ldots,\ f_n(b)$ entonces el número de raíces entre a y b está dado por $N(a)-N(b)$. Se supone que no hay raíces en a o en b. Si un punto elegido es una raíz de la función, este valor se puede grabar inmediatamente y evaluar la sucesión Sturm en un punto diferente.

Se puede presentar una complicación adicional si no hay residuo en algunos puntos en la sucesión de funciones. En este caso existe una raíz repetida, y ésta se determina por el último término de la sucesión $f_k(x)(k<n)$. Así, se puede usar la teoría de sucesión Sturm formando una nueva sucesión al dividir entre un factor común $f_k(x)$.

$$\tilde{f}_r(x)=\frac{f_r(x)}{f_k(x)} \qquad r=0,1,\ldots,k \tag{3.22}$$

El patrón de cambio de signo se puede usar para encontrar la posición de otras raíces.

3.4 Solución completa de un polinomio

Existen dos filosofías para la *solución completa de un polinomio*: ya sea encontrando de raíz en raíz, o bien encontrándolas todas al mismo tiempo. Ambas filosofías tienen sus propias características y poseen diferencias en cuanto a convergencia y crecimiento del error. Los métodos más comunes para encontrar las raíces en forma individual son:

1. Procedimiento de deflación
2. Método de Bairstow
3. Método de Laguerre

www.FreeLibros.me

4. Método de Bernoulli
5. Método de Newton

Por otra parte, los métodos más comunes que encuentran todo el conjunto de raíces en forma simultánea son:

6. Algoritmo de diferencia de cocientes
7. Método de Lehmer-Schur
8. Método de raíz cuadrada de Graeffe

A continuación se describe la forma operativa de cada uno de los métodos anteriores, tanto de los que encuentran las raíces individuales, como los que las encuentran en forma simultánea.

3.4.1 Procedimiento de deflación

Se sabe que un polinomio de grado n tiene n raíces, algunas de las cuales son reales y otras son complejas conjugadas. Las raíces complejas pueden, por tanto, representarse por un factor cuadrático con coeficientes reales. Esto es:

$$(x-\alpha-i\beta)(x-\alpha+i\beta)=x^2-2\alpha x+\alpha^2+\beta^2 \tag{3.23}$$

Primero, es necesario hacer estudios preliminares para encontrar aproximaciones a las raíces. Entonces se usa un método iterativo adecuado para encontrar el valor preciso de la raíz. Surge inmediatamente el problema de que los métodos iterativos no siempre convergen a la raíz deseada. En algunos casos, hay una raíz dominante; así, el proceso iterativo converge siempre a la misma raíz en un intervalo amplio de valores iniciales. En este caso, es necesario eliminar esta raíz para utilizar el método iterativo para encontrar las demás raíces. Esto se hace con el *método de deflación*. Se usa la división sintética, dividiendo entre el factor preciso, esto da como cociente un nuevo polinomio con una raíz menos. Si las raíces son complejas, es más conveniente usar el método iterativo de Bairstow para un factor cuadrático, ya que evita el uso de números complejos. Cuando se encuentra un factor cuadrático en forma precisa, el proceso de división sintética se puede usar nuevamente reduciendo el grado del polinomio en 2.

Desafortunadamente, esta sucesión de deflación e iteración puede generar grandes errores, que se pueden minimizar en dos formas. Primero, es posible encontrar las raíces en orden de magnitud creciente. Segundo, es posible invertir el polinomio y así encontrar las raíces en orden creciente de magnitud que corresponden al orden decreciente en magnitud del conjunto original. Si se tiene el polinomio

$$f(z)=a_0+a_1z+\cdots+a_{n-1}z^{n-1}+a_nz^n \tag{3.24}$$

se forma una nueva función aplicando la transformación $z=1/z$. Así se obtiene:

$$f(z)=a_0+\frac{a_1}{z}+\cdots+\frac{a_{n-1}}{z^{n-1}}+\frac{a_n}{z^n}=\frac{1}{z^n}\left(a_0z^n+a_1z^{n-1}+\cdots+a_{n-1}z+a_n\right) \tag{3.25}$$

El polinomio dentro de los paréntesis tiene raíces recíprocas de las raíces originales y ordenadas por amplitud en forma invertida. En el primer proceso, la raíz más pequeña es la más precisa, y en el segundo caso los inversos de las raíces más grandes serán más precisos.

EJEMPLO 3.5

Utilizando el procedimiento de deflación, encontrar las raíces del siguiente polinomio $f(z)=z^5+6z^4-6z^3-64z^2-27z+90$.

Solución. Utilizando el esquema de la ecuación (3.8) y tomando como primera raíz $\alpha_1=1$, se llega a:

www.FreeLibros.me

	1	6	−6	−64	−27	90
+1 +	0	1	7	1	−63	−90
	$p_5=1$	$p_4=7$	$p_3=1$	$p_2=-63$	$p_1=-90$	$p_0=0$

Así, la raíz es $z_1=1$, el residuo es $p_0=0$, y el cociente es $C_1(z)=z^4+7z^3+z^2-63z-90$. Para que el procedimiento de deflación dé como resultado un cociente que contiene las $n-1$ raíces restantes, es necesario que el proceso de división sintética llegue siempre a un valor de $p_0=0$. De otra forma la solución propuesta no es raíz del polinomio. Una vez que se obtiene una raíz, se puede repetir el proceso para obtener la siguiente raíz. Con $\alpha_2=-2$ se obtiene

	1	7	1	−63	−90
−2	0	−2	−10	18	90
	$p_4=1$	$p_3=5$	$p_2=-9$	$p_1=-45$	$p_0=0$

La nueva raíz es $z_2=-2$, y el nuevo cociente es $C_2(z)=z^3+5z^2-9z-45$. Aplicando nuevamente el procedimiento de división sintética con $\alpha_3=3$, se obtiene

	1	5	−9	−45
+3	0	3	24	45
	$p_3=1$	$p_2=8$	$p_1=15$	$p_0=0$

La nueva raíz es $z_3=3$ y el nuevo cociente es $C_3(z)=z^2+8z+15$. Aplicando de nuevo el procedimiento con $\alpha_4=-5$ se llega finalmente a,

	1	8	15
−5	0	−5	−15
	$p_2=1$	$p_1=3$	$p_0=0$

La nueva raíz es $z_4=-5$ y el nuevo cociente es $C_4(z)=z+3$. Por tanto, la última raíz es $z_5=-3$. Con esto se comprueba que utilizando la división sintética y el procedimiento de deflación se pueden encontrar una a una todas las raíces de un polinomio.

3.4.2 Método de Bairstow

Inicialmente, el método requiere de una aproximación $x^2+p_0x+q_0$ para un factor cuadrático. El proceso consiste en iterar para encontrar los incrementos Δp_r y Δq_r que mejoran la aproximación [Nakamura, 1992], [Maron *et al.*, 1995], [Nieves *et al.*, 2002], [Rodríguez, 2003]. La sucesión de valores;

$$p_{r+1}=p_r+\Delta p_r \qquad q_{r+1}=q_r+\Delta q_r \qquad r=0,1,2,\ldots, \tag{3.26}$$

se encuentra resolviendo las dos ecuaciones simultáneas

$$\begin{aligned} a_{11}\Delta p_r+a_{12}\Delta q_r&=b_1\\ a_{21}\Delta p_r+a_{22}\Delta q_r&=b_2 \end{aligned} \tag{3.27}$$

Los pasos que se siguen en este proceso son los siguientes:

1. El polinomio $f(x)$ se divide entre la aproximación a la ecuación cuadrática $x^2+p_rx+q_r$ para obtener un cociente $Q(x)$ y el residuo R_1x+S_1. Cuando el proceso converge R_1 y S_1 dan cero.
2. La función $xQ(x)$ se divide entonces entre $x^2+p_rx+q_r$. Esto da un nuevo residuo R_2x+S_2.
3. La función $Q(x)$ se divide entre $x^2+p_rx+q_r$. Esto da un residuo R_3x+S_3.

www.FreeLibros.me

4. Los coeficientes de las dos ecuaciones simultáneas están dados por: $a_{11} = -R_2$, $a_{12} = -R_3$, $a_{21} = -S_2$, $a_{22} = -S_3$, $b_1 = -R_1$ y $b_2 = -S_1$.
5. Se calculan los nuevos valores de p_{r+1} y q_{r+1}. El proceso se repite hasta que convergen los valores de p y q. Un rasgo del esquema que se ha ignorado, ya que no existe una solución simple, es el problema de encontrar la aproximación inicial del factor cuadrático. Una posibilidad es usar los tres términos de menor grado de la ecuación, aunque esto no garantiza la convergencia.

NOTA: La sección 3.7.3 proporciona el programa desarrollado en Matlab para el cálculo de un factor cuadrático por el método de Bairstow.

EJEMPLO 3.6

Utilizando el *método de Bairstow* aproximar un factor cuadrático del polinomio $f(x) = x^6 + 37x^5 + 520x^4 + 3\,490x^3 + 11\,449x^2 + 16\,633x + 8\,190$, con una aproximación inicial de $p_0 = 10$ y $q_0 = 40$.

SOLUCIÓN. Utilizando el método de división sintética entre un factor cuadrático $x^2 + p_0x + q_0$, se tiene que

−10	1	37	520	3 490	11 449	16 633	8 190
		−10	−270	−2 100	−3 100	510	−47 430
−40			−40	−1 080	−8 400	−12 400	2 040
	$p_6 = 1$	$p_5 = 27$	$p_4 = 210$	$p_3 = 310$	$p_2 = -51$	$p_1 = 4\,743$	$p_0 = -37\,200$

Por tanto $R_1 = 4\,743$ y $S_1 = -37\,200$. El cociente es $Q(x) = x^4 + 27x^3 + 210x^2 + 310x - 51$.
El siguiente paso es dividir $xQ(x)$ entre el factor cuadrático. Por tanto se tiene

−10	1	27	210	310	−51	0
		−10	−170	0	3 700	−36 490
−40			−40	−680	0	14 800
	$p_5 = 1$	$p_4 = 17$	$p_3 = 0$	$p_2 = -370$	$p_1 = 3\,649$	$p_0 = -21\,690$

Por tanto $R_2 = 3\,649$ y $S_2 = -21\,690$. El siguiente paso es dividir $Q(x)$ entre el factor cuadrático, así que

−10	1	27	210	310	−51
		−10	−170	0	3 700
−40			−40	−680	0
	$p_4 = 1$	$p_3 = 17$	$p_2 = 0$	$p_1 = -370$	$p_0 = 3\,649$

Por lo que $R_3 = -370$ y $S_3 = 3\,649$. Con estos valores se calculan los incrementos con la siguiente ecuación:

$$\begin{bmatrix} \Delta p \\ \Delta q \end{bmatrix} = \begin{bmatrix} -R_2 & -R_3 \\ -S_2 & -S_3 \end{bmatrix}^{-1} \begin{bmatrix} -R_1 \\ -S_1 \end{bmatrix}$$

Sustituyendo valores y realizando las operaciones matriciales, se llega a $\Delta p = 0.6698$ y $\Delta q = -6.2132$. Entonces, los nuevos valores del factor cuadrático son:

$$p_1 = p_0 + \Delta p = 10 + 0.6698 = 10.6698$$

$$q_1 = q_0 + \Delta q = 40 - 6.2132 = 33.7868$$

www.FreeLibros.me

Con estos nuevos valores se realizan las tres divisiones para dar nuevos incrementos. La tabla 3.2, de valores hasta llegar a converger a un factor cuadrático con un error máximo $\Delta p_n - \Delta p_{n-1} \leq 1e^{-3}$ y $\Delta q_n - \Delta q_{n-1} \leq 1e^{-3}$, es la siguiente:

Tabla 3.2 Resultados para obtener un factor cuadrático con el método de Bairstow.

	0	1	3	3	4	5	6
p	10.0000	10.6698	11.3011	11.8272	11.9915	12.0000	12.0000
q	40.0000	33.7868	32.5773	34.1730	34.9577	34.9999	35.0000

El factor cuadrático es, por tanto, $f(x) = x^2 + 12x + 35$, y el cociente que contiene las raíces restantes es $Q(x) = x^4 + 25x^3 + 185x^2 + 395x + 234$.

Del mismo modo, al cociente se le puede aplicar el método de Bairstow seguido del procedimiento de deflación para encontrar las raíces restantes.

3.4.3 Método de Laguerre

Suponiendo que todas las raíces de un polinomio $f(z)$ son reales y se pueden ordenar en forma ascendente como [Nieves *et al.*, 2002]:

$$\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_n$$

Si se define

$$I_i = [\alpha_i, \alpha_{i+1}] \text{ para } i = 1, 2, \ldots, n, \text{ con } \alpha_0 = -\infty \text{ y } \alpha_{n+1} = +\infty \tag{3.28}$$

Si se toma una aproximación arbitraria a una raíz de $f(z)$, ésta quedará dentro de un intervalo I_i. Así, la esencia del *método de Laguerre es construir una parábola con dos raíces reales* dentro del intervalo I_i; de esta forma al menos una de estas raíces estará más cercana a una raíz de $f(z)$ que a la aproximación dada. Por supuesto, se puede trazar un número infinito de parábolas, dependiendo del parámetro arbitrario de λ. Se elige el valor de λ de manera que una de las raíces de la parábola esté lo más cercana posible a una de las raíces de $f(z)$.

Inicialmente, para una valor real arbitrario de λ se tiene que,

$$S(\lambda) = \sum_{i=1}^{n} \left(\frac{\lambda - \alpha_i}{x - \alpha_i} \right)^2 > 0$$

por lo que la ecuación de la parábola será:

$$\varphi(y) = (x-y)^2 S(\lambda) - (\lambda - y)^2 \tag{3.29}$$

La ecuación (3.29) tiene dos raíces reales y distintas si $\lambda \pi \xi$, lo cual se acepta como un hecho. Si $f(x) \neq 0$, entonces $\varphi(x) < 0$ y $\varphi(\alpha_i) > 0$ para $i = 0, 1, 2, \ldots, n+1$. Por tanto, si $x \in I_i$, $i = 1, 2, \ldots, n$, las dos raíces de $\varphi(y)$ están en el intervalo I_i, una entre $[\alpha_i, x]$ y otra entre $[x, \alpha_{i+1}]$. Elegir λ de manera que una de las raíces de $\varphi(y)$ esté lo más cerca posible a una raíz de $f(z)$, significa maximizar $|x - y|$ como función de λ o alternativamente como una función del parámetro real $\mu = \lambda - x$. Así se tiene que:

$$\frac{f'(x)}{f(x)} = S_1 \equiv \sum_{i=1}^{n} \frac{1}{x - \alpha_i} \tag{3.30}$$

$$\frac{[f'(x)]^2 - f(x) f''(x)}{[f(x)]^2} = S_2 \equiv \sum_{i=1}^{n} \frac{1}{(x - \alpha_i)^2} \tag{3.31}$$

www.FreeLibros.me

Así se puede definir

$$\left(\frac{\lambda-\alpha_i}{x-\alpha_i}\right)^2=\frac{\mu^2}{(x-\alpha_i)^2}+\frac{2\mu}{x-\alpha_i}+1 \quad \textbf{(3.32)}$$

Utilizando las ecuaciones (3.30), (3.31) y (3.32), la ecuación de la parábola ahora se expresa como

$$\mu^2(\eta^2 S_2-1)+2\mu\eta(\eta S_1-1)+(n-1)\eta^2=0 \quad \textbf{(3.33)}$$

donde $\eta=x-y$. Analizando la ecuación (3.33), se puede notar que la solución de $\varphi(y)=0$ no depende de α_i. La ecuación (3.33) es una función cuadrática en μ, cuyas raíces son funciones continuas del parámetro η. Como μ sólo puede tomar valores reales, el objetivo es encontrar el máximo valor de $|\eta|$ que dé un valor real de μ. Para obtener este resultado se debe tener una valor de $|\eta|$ para el cual el discriminante de la ecuación cuadrática sea cero. Así se llega a la ecuación:

$$D=\eta^2\left(\left(S_1^2-(n-1)S_2\right)\eta^2-2\eta S_1+n\right) \quad \textbf{(3.34a)}$$

Resolviendo para $D=0$, se tiene que el valor de η es:

$$\eta=\frac{n}{S_1\pm\sqrt{(n-1)\left(nS_2-S_1^2\right)}} \quad \textbf{(3.34b)}$$

esto conduce a la ecuación,

$$y=x-\frac{n\,f(x)}{f'(x)\pm\sqrt{H(x)}} \quad \textbf{(3.35)}$$

donde

$$H(x)=(n-1)^2[f'(x)]^2-n(n-1)f(x)f''(x) \quad \textbf{(3.36)}$$

Cuando todas las raíces son reales, la función $H(x)$ siempre es un número real positivo. Así, en forma iterativa, se tiene el esquema

$$x_{i+1}=x_i-\frac{n\,f(x_i)}{f'(x_i)\pm\sqrt{H(x_i)}} \quad \textbf{(3.37)}$$

Para determinar el signo que se debe utilizar en el denominador, primero se determina el orden del polinomio, para lo cual se tiene que

$$F(x) \text{ tiene orden } p \text{ si y sólo si } \begin{array}{l} F(\alpha)=\alpha \\ F^{j}(\alpha)=0 \quad 1\le j<p \\ F^{p}(\alpha)\neq 0 \end{array}$$

De lo anterior se deduce que si se tiene que α es una raíz real simple de $f(x)$, entonces:

$$F(x)=x-\frac{n\,f(x)}{f'(x)\pm\sqrt{H(x)}}$$

$$F'(\alpha)=1-\frac{n\,f'(\alpha)}{f'(\alpha)\pm\sqrt{H(\alpha)}}=1-\frac{n\,f'(\alpha)}{f'(\alpha)\pm(n-1)|f'(\alpha)|}$$

$$F''(\alpha)=\frac{-n\,f''(\alpha)}{f'(\alpha)\pm(n-1)|f'(\alpha)|}\times\left(1-\frac{2f'(\alpha)}{f'(\alpha)\pm(n-1)|f'(\alpha)|}\left(1\pm\frac{n-2f'(\alpha)}{2|f'(\alpha)|}\right)\right)$$

www.FreeLibros.me

De las expresiones anteriores se puede notar que si el signo de la expresión $\sqrt{H(x)}$ se escoge de manera que sea igual al signo de $f'(\alpha)$, entonces, tanto $F'(\alpha)$ como $F''(\alpha)$ serán cero. Por tanto, en la práctica se elige el signo de $\sqrt{H(x)}$ igual al signo de $f'(x_i)$. De esta forma se puede decir que el método de Laguerre es de tercer orden para raíces reales simples. El método funciona a expensas de calcular $f(x_i)$, $f'(x_i)$ y $f''(x_i)$ en cada iteración. En la sección 3.7.4 se proporciona el código Matlab del método de Laguerre para el cálculo de raíces reales simples.

EJEMPLO 3.7

Utilizando el *método de Laguerre*, calcular una de las raíces del siguiente polinomio: $x^6+45x^5+802x^4+7\,236x^3+34\,792x^2+84\,384x+80\,640=0$. Iniciar con $x_0=120$.

Solución. Primero se evalúa la función y sus primeras dos derivadas en $x_0=120$. Así se obtiene

$$f(x_0)=4.285045(10)^{12}$$

$$f'(x_0)=2.018196(10)^{11}$$

$$f''(x_0)=7.919865(10)^{9}$$

Se evalúa el *polinomio de Laguerre*

$$H(x_0)=1.697884(10)^{20}$$

Se calcula la siguiente aproximación tomando $f'(x_0)$ y $H(x_0)$ con el mismo signo.

$$x_1=0.333809$$

Los resultados completos de las iteraciones se muestran en la tabla 3.3.

Tabla 3.3 Resultados de la aplicación del método de Laguerre.

x_0	x_1	x_2	x_3	x_4	x_5
120	0.333809	−2.597270	−2.990968	−2.999999	−3.0

La raíz de menor módulo del polinomio es efectivamente $x=-3$. Analizando la tabla anterior se puede notar cómo el método de Laguerre converge rápidamente a la raíz de menor módulo, aun cuando la aproximación inicial sea muy burda. Por lo anterior se concluye que en el caso de tener un polinomio con raíces reales, el método de Laguerre es una buena opción de cálculo.

3.4.4 Método de Bernoulli

Si se tiene la función

$$f(z)=a_nz^n+a_{n-1}z^{n-1}+\cdots+a_1z+a_0 \tag{3.38}$$

y además se considera la ecuación en diferencias con los mismos coeficientes

$$a_nu_k+a_{n-1}u_{k-1}+\cdots+a_0u_{k-n}=0, \tag{3.39}$$

si las raíces α_i de la ecuación (3.38) son reales y distintas, la solución de la ecuación (3.39) está dada por

www.FreeLibros.me

$$u_k = \sum_{i=1}^{n} c_i \, \alpha_i^{\,k} \tag{3.40}$$

donde los c_i dependen de las condiciones usadas para resolver la ecuación (3.39).

Si las raíces se ordenan en magnitud decreciente, la ecuación (3.40) se puede escribir como:

$$u_k = c_1 \, \alpha_1^{\,k} \left[1 + \sum_{i=2}^{n} \frac{c_i}{c_1} \left(\frac{\alpha_i}{\alpha_1} \right)^k \right] \tag{3.41}$$

De aquí se deduce que si $c_1 \neq 0$, entonces

$$\lim_{k \to \infty} \frac{u_k}{u_{k-1}} = \alpha_1 \tag{3.42}$$

La esencia del método de Bernoulli es usar la ecuación (3.39) para generar los valores sucesivos de u_k y después calcular el radio de los valores sucesivos de u_k hasta la convergencia.

Los valores iniciales para usarse en la ecuación (3.39) se generan con la siguiente ecuación:

$$a_n u_m + a_{n-1} u_{m-1} + \cdots + a_{n-m+1} u_1 + m a_{n-m} = 0 \tag{3.43}$$

donde $m = 1, 2, \ldots, n$.

En la sección 3.7.5 se proporciona el código Matlab del método de Bernoulli para calcular las raíces reales simples.

EJEMPLO 3.8

Utilizando el *método de Bernoulli*, calcular la raíz de mayor módulo del siguiente polinomio: $f(x) = x^5 + 17x^4 + 105x^3 + 295x^2 + 374x + 168$.

Solución. Primero con la ecuación (3.43) se generan las condiciones iniciales, así se obtiene:

Con $m = 1$ se tiene $\quad a_5 u_1 + a_4 = 0$

Con $m = 2$ se tiene $\quad a_5 u_2 + a_4 u_1 + 2a_3 = 0$

Con $m = 3$ se tiene $\quad a_5 u_3 + a_4 u_2 + a_3 u_1 + 3a_2 = 0$

Con $m = 4$ se tiene $\quad a_5 u_4 + a_4 u_3 + a_3 u_2 + a_2 u_1 + 4a_1 = 0$

Con $m = 5$ se tiene $\quad a_5 u_5 + a_4 u_4 + a_3 u_3 + a_2 u_2 + a_1 u_1 + 5a_0 = 0$

Sustituyendo los valores de los coeficientes dados por el polinomio $f(x)$ se genera el siguiente grupo de ecuaciones,

$$u_1 + 17 = 0$$

$$u_2 + 17u_1 + 210 = 0$$

$$u_3 + 17u_2 + 105u_1 + 885 = 0$$

$$u_4 + 17u_3 + 105u_2 + 295u_1 + 1\,496 = 0$$

$$u_5 + 17u_4 + 105u_3 + 295u_2 + 374u_1 + 840 = 0$$

www.FreeLibros.me

En forma matricial, se tiene que:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 17 & 1 & 0 & 0 & 0 \\ 105 & 17 & 1 & 0 & 0 \\ 295 & 105 & 17 & 1 & 0 \\ 374 & 295 & 105 & 17 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} = \begin{bmatrix} -17 \\ -210 \\ -885 \\ -1\,496 \\ -840 \end{bmatrix}$$

De este grupo de ecuaciones se obtienen los valores iniciales para utilizar en la ecuación (3.39), los cuales son: $u_1 = 17$, $u_2 = 79$, $u_3 = -443$, $u_4 = 2\,775$ y $u_5 = -18\,107$.

La ecuación en diferencias queda de la siguiente manera:

$$u_k = -17u_{k-1} - 105u_{k-2} - 295u_{k-3} - 374u_{k-4} - 168u_{k-5}$$

Iniciando en $k = 6$, se obtiene,

$$u_6 = 122\,539$$

Así, de acuerdo a la fórmula (3.42), la primera aproximación a la raíz es:

$$\text{Raíz} = \frac{u_6}{u_5} = -6.767493$$

La tabla 3.4 muestra los resultados hasta obtener dos valores consecutivos de la raíz que difieren en menos de $1(10)^{-3}$.

Tabla 3.4 Resultados de la aplicación del método de Bernoulli.

Coeficiente	*Raíz*
$u_6 = 1.225389 \times 10^5$	$(u_6/u_5) = -6.767493$
$u_7 = -8.422429 \times 10^5$	$(u_7/u_6) = -6.873264$
$u_8 = 5.837154 \times 10^6$	$(u_8/u_7) = -6.930487$
$u_9 = -4.063594 \times 10^7$	$(u_9/u_8) = -6.961601$
$u_{10} = 2.835838 \times 10^8$	$(u_{10}/u_9) = -6.978646$
$u_{11} = -1.981700 \times 10^9$	$(u_{11}/u_{10}) = -6.988056$
$u_{12} = 1.385859 \times 10^{10}$	$(u_{12}/u_{11}) = -6.993287$
$u_{13} = -9.695772 \times 10^{10}$	$(u_{13}/u_{12}) = -6.996213$
$u_{14} = 6.784963 \times 10^{11}$	$(u_{14}/u_{13}) = -6.997857$
$u_{15} = -4.748649 \times 10^{12}$	$(u_{15}/u_{14}) = -6.998784$
$u_{16} = 3.323726 \times 10^{13}$	$(u_{16}/u_{15}) = -6.999309$

Analizando los resultados del ejercicio, se puede notar que el método de Bernoulli es de convergencia lenta. Sin embargo, se puede apreciar que la primera aproximación es muy buena y no necesita condiciones iniciales externas; es decir, el método por sí mismo fija las condiciones iniciales. Por esta razón, aunque es un método de convergencia lenta, es bastante valioso como primera aproximación para otro método de convergencia rápida.

www.FreeLibros.me

3.4.5 Método de Newton

El objetivo de este método es encontrar la raíz de menor módulo de un polinomio dado. De esta forma, el procedimiento de deflación se da en forma estable, y así se encuentran las demás raíces [Cordero *et al.*, 2006]. Dado un polinomio $f(z)$ de la forma

$$f(z) = a_n z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0, \tag{3.44}$$

se desea generar la secuencia z_k que converge a la raíz de menor magnitud, o al menos, a una muy cercana a ésta. Los puntos sucesivos se relacionan con la fórmula,

$$z_{k+1} = z_k + m dz_k \tag{3.45}$$

donde m es un escalar.

Un paso adecuado es aquel donde $z_{k+1} \neq z_k$. Si la iteración cumple con esta diferencia, la corrección de Newton es:

$$w_k = -\frac{f(z_k)}{f'(z_k)} \tag{3.46}$$

De esta forma se calcula el valor de dz_k como,

$$dz_k = \begin{cases} w_k & \text{si se cumple que} \quad |w_k| \leq 3|z_k - z_{k-1}| \\ \dfrac{3|z_k - z_{k-1}| e^{i\phi} w_k}{|w_k|} & \text{en cualquier otra situación} \end{cases} \tag{3.47}$$

donde el ángulo ϕ se escoge de manera arbitraria como $\phi = 45°$. Si el paso previo no es adecuado entonces se tiene que,

$$dz_k = -\tfrac{1}{2} e^{i\phi} dz_{k-1} \tag{3.48}$$

El razonamiento detrás de esta elección es el siguiente:

1. Si el paso es adecuado, pero el valor absoluto de $|w_k|$ es relativamente grande, la función se está aproximando a un punto donde $f'(z) \to 0$ y, por tanto, se cambia la dirección de búsqueda un ángulo ϕ.
2. Si el paso no es adecuado, se desea cambiar la dirección para encontrar una iteración adecuada. Pero si $f(z) \neq 0$, existe una dirección descendente para la función, ya que $f(z)$ tiene un mínimo local (o global) sólo en las raíces, por lo que buscando en diferentes direcciones conducirá a una dirección descendente.

El *método de Newton* consta de dos etapas. En la primera se aproxima la raíz lo más posible a la raíz del polinomio, al punto que se pueda garantizar la convergencia. Una vez que se garantiza que el método converge, se llega a la etapa dos, donde se utiliza la fórmula estándar de Newton para calcular la raíz. Para garantizar la convergencia del método de Newton iniciando con z_k, se debe de cumplir la siguiente desigualdad:

$$2|f(z_k)| \cdot |f'(z_{k-1}) - f'(z_k)| \leq |f'(z_k)|^2 \cdot |z_{k-1} - z_k| \tag{3.49}$$

Esta prueba se debe hacer en cada iteración de la etapa dos. En el momento en que se deje de cumplir, se retrocede a la primera etapa.

Una vez que se elige dz_k, se calcula $F(z_k + dz_k)$ y se prueba la desigualdad

$$F(z_k + dz_k) < F(z_k) \tag{3.50}$$

www.FreeLibros.me

Si se cumple la desigualdad, se calcula $F(z_k + mdz_k)$ con $m = 2, 3, \ldots, n$. Si la desigualdad no se cumple, se calcula la función con $m = \left[\frac{1}{2} \quad \frac{1}{4} \quad \frac{1}{4}e^{i\phi}\right]$.

Si $F(z_k + dz_k) \geq F(z_k)$ y $F(z_k + \frac{1}{2}dz_k) \geq F(z_k + dz_k)$, entonces se toma $m = 0$ para tener un paso no adecuado y cambiar la dirección de búsqueda.

La primera etapa se inicia con las siguientes condiciones

$z_0 = 0$

$$dz_k = \begin{cases} -\dfrac{f(0)}{f'(0)} & \text{si se cumple que } f'(0) \neq 0 \\ 1 & \text{en cualquier otra situación} \end{cases}$$

$$z_1 = \frac{1}{2} \min_{k>0} \left(\left| \frac{a_0}{a_k} \right|^{\frac{1}{k}} \right) \frac{dz_0}{|dz_0|}$$

La elección de z_1 es tal que la magnitud sea menor que cualquier raíz de $f(z)$. El código del método se proporciona en la sección 3.7.6 de este mismo capítulo. El programa está desarrollado en la plataforma de Matlab y generalizado para calcular las raíces del polinomio.

EJEMPLO 3.9

Utilizando el método de Newton calcular la raíz de menor módulo del siguiente polinomio $f(z) = z^7 + 28z^6 + 322z^5 + 1\,960z^4 + 6\,769z^3 + 13\,132z^2 + 13\,068z + 504$.

Solución. Primero se fijan las condiciones iniciales del método como:

$$z_0 = 0\,,\; dz_0 = -0.0386 \text{ y } z_1 = -0.0002$$

La primera etapa se inicia cuando se cumple con la condición de convergencia. La tabla 3.4 resume los resultados de esta etapa.

Tabla 3.4 Resultados de la primera etapa del método de Newton.

z_k	dz_k
$z_0 = 0$	$dz_0 = -0.0386$
$z_1 = -0.0002$	$dz_1 = -0.00004 - 0.0004i$
$z_2 = -0.0033 - 0.0031i$	$dz_2 = -0.0099 - 0.0084i$
$z_3 = -0.0231 - 0.0199i$	$dz_3 = -0.0172 + 0.0192i$
$z_4 = -0.0574 + 0.0186i$	$dz_4 = -0.0344 + 0.0385i$

Una vez que se verifica que el método converge, se pasa a la segunda etapa. La tabla 3.5 muestra los resultados de esta segunda etapa.

Tabla 3.5 Resultados de la segunda etapa del método de Newton.

z_k	dz_k
$z_5 = -0.0402 - 0.0007i$	$dz_5 = 0.0172 - 0.0192i$
$z_6 = -0.0402 + 0.0000i$	$dz_6 = 0.0001 + 0.0007i$
$z_7 = -0.0402 - 0.0000i$	$dz_7 = 0.0000 - 0.0000i$

El método se detiene cuando dz_7 es menor que una tolerancia especificada. En este caso fue de $1(10)^{-4}$. Con esto se llega a la raíz $z_k = -0.0402$, la cual es efectivamente la raíz de menor módulo.

www.FreeLibros.me

3.4.6 Algoritmo de diferencia de cocientes

Se puntualizó el problema del crecimiento del error que se da cuando se usa un proceso iterativo, seguido de deflación, repetidamente. Entonces resulta atractivo encontrar todas las raíces de un polinomio en forma simultánea. Desafortunadamente, los método de este tipo tienen la desventaja de que, cuando los resultados están en la vecindad de los valores correctos, la rapidez de convergencia es, en general, mucho menor que la que se logra con un buen método de raíces simples. Una buena estrategia es usar un método que encuentra todas las raíces para dar una buena aproximación inicial, combinado con un método de convergencia eficaz para mejorar la precisión de cada raíz individualmente.

En el algoritmo de *diferencia de cocientes,* primero se construye una tabla de $2n+1$ columnas para un polinomio de grado n. Las dos de los extremos son ceros. Las columnas restantes están en dos conjuntos alternados, las cuales se obtienen de dos reglas de cálculo diferentes, la regla de cocientes y la regla de diferencias. El método de generación de la tabla es más fácil de explicar mediante su construcción partiendo de la columna izquierda de ceros, basándose en el método de Bernoulli para encontrar raíces. Sin embargo, para reducir los errores en la construcción, la tabla se genera renglón por renglón. Esto necesita un método para generar los primeros dos renglones. La tabla 3.6 muestra el ejemplo para una función cúbica. En la tabla los superíndices son constantes para cada columna y los subíndices son constantes a lo largo de la diagonal hacia delante.

Tabla 3.6 Tabla de diferencia de cocientes.

	$q_0^{(1)}$		$q_{-1}^{(2)}$		$q_{-2}^{(3)}$	
0		$\varepsilon_0^{(1)}$		$\varepsilon_{-1}^{(2)}$		0
	$q_1^{(1)}$		$q_0^{(2)}$		$q_{-1}^{(3)}$	
0		$\varepsilon_1^{(1)}$		$\varepsilon_0^{(2)}$		0
	$q_2^{(1)}$		$q_1^{(2)}$		$q_0^{(3)}$	
0		$\varepsilon_2^{(1)}$		$\varepsilon_1^{(2)}$		0
	$q_3^{(1)}$		$q_2^{(2)}$		$q_1^{(3)}$	
$\vdots$		$\vdots$		$\vdots$		$\vdots$

Para un esquema de grado n el proceso se inicia calculando los elementos de los dos primeros renglones como sigue:

$$q_0^{(1)} = -\frac{a_{n-1}}{a_n} \tag{3.51a}$$

$$q_{1-r}^{(r)} = 0 \qquad r = 2, 3, \ldots, n \tag{3.51b}$$

$$\varepsilon_{1-r}^{(r)} = \frac{a_{r-1}}{a_r} \qquad r = n-1, n-2, \ldots, 1 \tag{3.51c}$$

Los elementos se calculan moviéndose a la derecha a lo largo de los renglones, y hacia abajo a lo largo de las columnas usando los cuatro valores en los puntos del rombo como sigue:

$$\begin{matrix} & \alpha_j & \\ \beta_{j+1} & & \beta_j \\ & \alpha_{j+1} & \end{matrix} \tag{3.52}$$

Si los elementos α_j quedan en una columna q entonces $\alpha_j + \beta_j = \alpha_{j+1} + \beta_{j+1}$. Si los elementos α_j quedan en una columna ε entonces $\alpha_j \times \beta_j = \alpha_{j+1} \times \beta_{j+1}$. Resolviendo renglón por renglón, se conocen las cantidades α_j, β_j y β_{j+1} y los valores desconocidos de α_{j+1} se encuentran con una de las siguientes dos ecuaciones:

www.FreeLibros.me

$$q_{j+1}^{(r)} = q_j^{(r)} + \varepsilon_j^{(r)} - \varepsilon_{j+1}^{(r-1)} \tag{3.53a}$$

$$\varepsilon_{j+1}^{(r)} = \varepsilon_j^{(r)} \frac{q_j^{(r+1)}}{q_{j+1}^{(r)}} \tag{3.53b}$$

Este esquema termina si cualquier valor de $q_j^{(r)}$ se hace cero, ya que, como se sabe, no se puede dividir entre cero. El algoritmo anterior es más útil cuando las raíces $z_r\,(r=1, 2,\ldots, n)$ satisfacen la relación $|z_1|>|z_2|>\cdots>|z_n|$. En este caso se puede probar que:

$$\lim_{j\to\infty} q_j^{(r)} = z_r \qquad r=1, 2, 3,\ldots, n \tag{3.54a}$$

$$\lim_{j\to\infty} \varepsilon_j^{(r)} = 0 \qquad r=1, 2, 3,\ldots, n \tag{3.54b}$$

Así, el esquema revela si todas las raíces son distintas en módulo analizando el comportamiento de las columnas ε. Si algunas de las raíces son de igual módulo, por ejemplo, $|z_1|\ge|z_2|\ge\cdots\ge|z_n|$, entonces se tienen las siguientes condiciones:

$$\text{Para toda } r \text{ tal que } |z_{r-1}|>|z_r|>|z_{r+1}|,\ \lim_{j\to\infty} q_j^{(r)} = z_r \tag{3.55a}$$

$$\text{Para toda } r \text{ tal que } |z_r|>|z_{r+1}|,\ \lim_{j\to\infty} \varepsilon_j^{(r)} = 0 \tag{3.55b}$$

Así, la tabla se divide en grupos de columnas q, correspondientes a raíces de igual módulo, y columnas ε que tienden a cero.

El caso más común de raíces de igual módulo ocurre cuando una ecuación con coeficientes reales tiene raíces complejas conjugadas. Se tendrán aquí dos columnas q separadas por una columna ε que no tiende a cero y las dos raíces, z_{r+1} y z_{r+2}, que son la solución de la ecuación cuadrática,

$$z^2 + A_r z + B_r = 0$$

donde

$$\begin{aligned} \lim_{j\to\infty}\left(q_{j+1}^{(r+1)} + q_j^{(r+2)}\right) &= A_r \\ \lim_{j\to\infty}\left(q_j^{(r+1)} + q_j^{(r+2)}\right) &= B_r \end{aligned} \tag{3.56}$$

En la sección 3.7.7 se proporciona un código creado por Matlab, donde se desarrolló este algoritmo. El código es general para el caso de raíces reales simples.

EJEMPLO 3.10

Utilizando el método de *diferencia de cocientes*, aproximar todas las raíces del siguiente polinomio de cuarto orden; $f(z) = z^4 + 17z^3 + 101z^2 + 247z + 210$.

SOLUCIÓN. Con este polinomio se construye la tabla 3.7 de nueve columnas, de la forma que se muestra en la página 78.

Analizando la tabla 3.7 se puede notar cómo las ε tienden a cero. En este caso específico, el proceso se detuvo cuando el mayor valor de ε es de una milésima. Igualmente se ve cómo en las columnas de q quedan acomodadas las raíces de mayor a menor módulo. En este caso, las raíces exactas son $z_1=-7$, $z_2=-5$, $z_3=-3$ y $z_4=-2$, por lo que, aun cuando el método converge con lentitud, desde las primeras iteraciones se puede notar cómo el método se dirige a las raíces en forma estable.

www.FreeLibros.me

Tabla 3.7 Tabla de resultados al aplicar el algoritmo de diferencia de cocientes.

	q^1	ε^1	q^2	ε^2	q^3	ε^3	q^4	
	−17		0		0		0	
0		5.9412		2.4455		0.8502		0
	−11.0588		−3.4956		−1.5953		−0.8502	
0		1.8780		1.1161		0.4531		0
	−9.1809		−4.2575		−2.2583		−1.3033	
0		0.8709		0.5920		0.2615		0
	−8.3100		−4.5364		−2.5889		−1.5648	
0		0.4754		0.3379		0.1580		0
	−7.8346		−4.6739		−2.7687		−1.7228	
0		0.2836		0.2001		0.0983		0
	−7.5509		−4.7574		−2.8705		−1.8212	
0		0.1787		0.1208		0.0624		0
	−7.3722		−4.8153		−2.9289		−1.8836	
0		0.1167		0.0735		0.0401		0
	−7.2555		−4.8586		−2.9622		−1.9237	
0		0.0782		0.0448		0.0261		0
	−7.1774		−4.8920		−2.9809		−1.9498	
0		0.0533		0.0273		0.0170		0
	−7.1241		−4.9179		−2.9912		−1.9668	
0		0.0368		0.0166		0.0112		0
	−7.0873		−4.9381		−2.9966		−1.9780	
0		0.0256		0.0101		0.0074		0
	−7.0617		−4.9537		−2.9992		−1.9854	
0		0.0180		0.0061		0.0049		0
	−7.0437		−4.9655		−3.0004		−1.9903	
0		0.0127		0.0037		0.0032		0
	−7.0311		−4.9745		−3.0009		−1.9935	
0		0.0090		0.0022		0.0022		0
	−7.0221		−4.9813		−3.0009		−1.9957	
0		0.0064		0.0013		0.0014		0
	−7.0157		−4.9863		−3.0008		−1.9971	
0		0.0045		0.0008		0.0010		0
	−7.0112		−4.9900		−3.0007		−1.9981	
0		0.0032		0.0005		0.0006		0
	−7.0080		−4.9927		−3.0005		−1.9987	
0		0.0023		0.0003		0.0004		0
	−7.0057		−4.9947		−3.0004		−1.9992	
0		0.0016		0.0002		0.0003		0
	−7.0041		−4.9962		−3.0003		−1.9944	
0		0.0012		0.0001		0.0002		0

www.FreeLibros.me

3.4.7 Método de Lehmer-Schur

Aunque este método siempre converge, es extremadamente laborioso y muy lento. Por supuesto, es adecuado para encontrar la primera aproximación. El *método Lehmer-Schur* se basa en una sucesión de cálculos que verifica si una raíz queda dentro de un círculo de prueba. Se produce una sucesión de círculos de prueba que se puede usar para buscar en todo el plano complejo, y que va a converger a una raíz cuando ésta se aísla (véase la figura 3.1). Primero, la prueba se usa para determinar si existe una raíz dentro de un círculo con centro en el origen y de módulo unitario. Si la prueba no da resultado, se transforma la ecuación original para buscar en un círculo con el doble de radio.

Si continúa sin dar resultados se hace una transformación adicional con un círculo de radio cuatro veces mayor, y así sucesivamente. Este proceso se repite hasta que se encuentra una raíz que quede dentro del intervalo $R<|z|<2R$.

El área en la cual está la raíz se estrecha buscando en 8 círculos sobrepuestos de radio $4R/5$ con centro en los puntos.

$$z=\frac{3Re^{i2\pi k/8}}{2\cos(\pi/8)} \qquad k=0,1,2,\ldots,7 \tag{3.57}$$

Uno de estos círculos puede contener una raíz, aunque es importante resaltar el hecho de que más de un círculo puede contener una raíz, ya que los círculos sobrepuestos cubren toda el área del intervalo y parte de esta área en forma doble. Teniendo un círculo que contenga una raíz, se hace la búsqueda en una sucesión de círculos, ahora de la mitad de radio, que dará un intervalo que contiene la raíz y luego se vuelve a repetir el proceso.

3.4.8 Método de raíz cuadrada de Graeffe

En su forma más sencilla, este método toma una ecuación que tiene raíces distintas en magnitud y de ésta se obtiene una segunda ecuación con raíces que son el cuadrado de las raíces originales. El proceso se repite hasta que las nuevas raíces son 2^m veces las raíces originales. Para una m muy grande, la magnitud de las raíces originales se puede determinar de una forma muy fácil. El método es atractivo porque el proceso de elevación al cuadrado incrementa la separación de las raíces, lo cual simplifica el proceso de identificación de las raíces. Si se tiene un polinomio de la forma:

$$f(z)=a_nz^n+a_{n-1}z^{n-1}+\cdots+a_1z+a_0, \tag{3.58}$$

si se supone que $a_n=1$, entonces se puede hacer

$$f_0(z)=(z-\alpha_1)(z-\alpha_2)\cdots(z-\alpha_n) \tag{3.59}$$

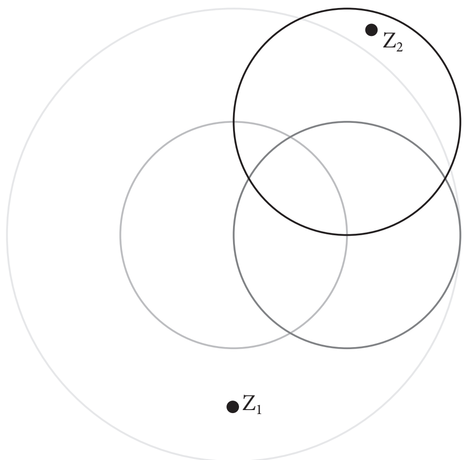


Figura 3.1 Método de Lehmer-Schur.

www.FreeLibros.me

Si se hace $w = z^2$, se puede obtener

$$f_1(w) = (-1)^n f_0(z) f_0(-z) = (w - \alpha_1^2)(w - \alpha_2^2)\cdots(w - \alpha_n^2) \tag{3.60}$$

de manera que las raíces de $f_1(w)$ son los cuadrados de las raíces de $f_0(z)$; así se forma la secuencia

$$f_{r+1}(w) = (-1)^n f_r(z) f_r(-z), \text{ para } r = 0, 1, \ldots \tag{3.61}$$

En este caso, las raíces de un polinomio son los cuadrados de las raíces del polinomio anterior. Si los coeficientes de los polinomios obtenidos con este método se denotan como $a_r^{(j)}$ donde $a_r^{(0)} = a_r$, como se define en la ecuación (3.58), los coeficientes de los nuevos polinomios se obtienen como sigue:

$$a_n^{(1)} = \left(a_n^{(0)}\right)^2 \tag{3.62a}$$

$$a_{n-1}^{(1)} = -\left(a_{n-1}^{(0)}\right)^2 + 2a_n^{(0)}a_{n-2}^{(0)} \tag{3.62b}$$

$$a_{n-2}^{(1)} = \left(a_{n-2}^{(0)}\right)^2 - 2a_{n-1}^{(0)}a_{n-3}^{(0)} + 2a_n^{(0)}a_{n-4}^{(0)} \tag{3.62c}$$

En forma general, se tiene que:

$$a_j^{(r+1)} = (-1)^{n-j}\left[\left(a_j^{(r)}\right)^2 + 2\sum_{k=1}^{\text{mín}[n-j\;\;j]} (-1)^k a_{j-k}^{(r)} a_{j+k}^{(r)}\right] \tag{3.63}$$

Para utilizar la sucesión de polinomios $f_r(z)$ se necesita la ecuación que relaciona los coeficientes de un polinomio con sus raíces, es decir

$$a_j^{(r)} = (-1)^{n-j} S_{n-j}\left(\alpha_1^{2^r}, \alpha_2^{2^r}, \ldots, \alpha_n^{2^r}\right), \text{ para } j = 0, 1, \ldots, n-1 \tag{3.64}$$

donde

$$S_k(x_1, x_2, \ldots, x_n) = \sum_{k=1}^{n}{}_C\; x_{r1}x_{r2}\cdots x_{rk}$$

donde $\sum_{k=1}^{n}{}_C$ indica que la suma se realiza tomando en cuenta todas las combinaciones de k, desde 1 hasta n. Así, por ejemplo

$$a_{n-1}^{(r)} = -S_1\left(\alpha_1^{2^r}, \alpha_2^{2^r}, \ldots, \alpha_n^{2^r}\right) = -\sum_{k=1}^{n} \alpha_k^{2^r} \tag{3.65}$$

lo cual lleva a la expresión:

$$a_{n-1}^{(r)} = -\alpha_1^{2^r}\left[1 + \sum_{k=2}^{n}\left(\frac{\alpha_k}{\alpha_1}\right)^{2^r}\right] \tag{3.66}$$

Si se tiene que las raíces son diferentes en magnitud y están ordenadas de manera que

$$\rho_1 > \rho_2 > \cdots > \rho_n, \tag{3.67}$$

entonces se tiene

$$\lim_{r\to\infty}\left|a_{n-1}^{(r)}\right|^{\frac{1}{2^r}} = |\alpha_1| \tag{3.68}$$

Por tanto, para una r suficientemente grande, se tiene

$$\rho_1 \approx \left|a_{n-1}^{(r)}\right|^{\frac{1}{2^r}} \tag{3.69}$$

En forma similar se tiene que

$$a_{n-2}^{(r)} = \alpha_1^{2^r}\alpha_2^{2^r}\left[1 + \sum_{k=2}^{n}\left(\frac{\alpha_k}{\alpha_1\alpha_2}\right)^{2^r}\right] \tag{3.70}$$

www.FreeLibros.me

y en consecuencia, para una r suficientemente grande, se tiene

$$\rho_2 \approx \frac{1}{\rho_1}\left|a_{n-2}^{(r)}\right|^{\frac{1}{2^r}} \tag{3.71}$$

Generalizando la fórmula anterior se llega a

$$\rho_k \approx \frac{1}{\rho_1 \cdots \rho_{k-1}}\left|a_{n-k}^{(r)}\right|^{\frac{1}{2^r}} \text{ para } k = 2, 3, \ldots, n \tag{3.72}$$

En la práctica una r suficientemente grande significa que el *método de la raíz cuadrada de Graeffe* continúa hasta tener una aproximación que se estabiliza en la cantidad de decimales que se desean. El código en Matlab de este método se presenta en la sección 3.7.8.

EJEMPLO 3.11

Utilizando el *método de raíz cuadrada de Graeffe*, aproximar todas las raíces del siguiente polinomio de tercer orden; $f(z) = z^3 + 26z^2 + 203z + 490$.

SOLUCIÓN. Utilizando la fórmula (3.63), se calculan los coeficientes de cada iteración, y las raíces se calculan con las fórmulas (3.69) para la primera y (3.72) para las siguientes.

La tabla 3.8 muestra loa resultados de aplicar el método en cinco iteraciones.

Tabla 3.8 Tabla de resultados de aplicar el método de la raíz cuadrada de Graeffe.

r	1	2	3	4	5
a_3	1	1	1	1	1
a_2	26	-270	-41442	$-1.481944482 \times 10^{9}$	$-2.177986723327 \times 10^{18}$
a_1	203	15729	117747441	$9.086362201208 \times 10^{15}$	$7.271210643588 \times 10^{31}$
a_0	490	-240100	-5.764801×10^{10}	$-3.32329305696 \times 10^{21}$	$-1.104427674243 \times 10^{43}$
ρ_1	16.4317	14.2679	14.0073	14.0000	14.0000
ρ_2	7.6325	7.3009	7.0542	7.0020	7.0000
ρ_3	3.9070	4.7039	4.9590	4.9986	5.0000

En este caso se iteró hasta que se tuvo un error de $1(10)^{-3}$, lo cual se logró en cinco iteraciones. Analizando la tabla de resultados, se puede notar que los coeficientes son muy grandes. Por esta razón no se puede hacer el número de iteraciones que se quieran. Asimismo, analizando las raíces, se puede notar que en cada iteración siempre se obtiene signo positivo. Dicho de otra forma, el método en realidad sólo aproxima el módulo de la raíz. El signo se puede obtener de manera sencilla con el procedimiento de división sintética. Por las razones anteriores, este método es adecuado para hacer una primera aproximación de los módulos de todas las raíces de un polinomio y después se puede utilizar un método que no tenga las limitantes antes mencionadas.

3.5 Método de Jenkins-Traub

Considerando un polinomio mónico de la forma:

$$P(z) = z^n + a_{n-1}z^{n-1} + \cdots + a_1 z + a_0 \tag{3.73}$$

donde los a_i son números complejos con $a_0 \neq 0$, se puede generar una secuencia de polinomios $H^k(z)$, iniciando con $H^0(z) = P'(z)$, todos de la forma

www.FreeLibros.me

$$H^{k+1}(z)=\sum_{j=2}^{n}\left[Q^{k}(j)+T\cdot Q^{k}(j-1)\right] \tag{3.74}$$

donde

$$H^{k+1}(1)=P(1)$$

$$T^{k}=-\frac{P(s)}{H^{k}(s)}$$

$$Q^{k}(j)=Q^{k}(j-1)\cdot S_{k}+P(j),\text{ para } j=2,\ldots,n;\text{ con } Q^{k}(1)=P(1)$$

Así, se puede aproximar la raíz α_1 con la fórmula (lo que equivale al método de Newton):

$$S_{k+1}=S_{k}-\frac{P(s)}{H^{k}(s)} \tag{3.75}$$

Como un estimado inicial de S_k, se elige

$$S_0=\beta\cdot e^{i\theta} \tag{3.76}$$

donde θ es un ángulo arbitrario y β es la única raíz positiva del polinomio,

$$z^{n}+|a_{n-1}|z^{n-1}+\cdots+|a_1|z-|a_0| \tag{3.77}$$

El teorema de Cauchy garantiza que β es el límite inferior de $|\alpha_j|$, $j=1,\ldots,p$.

3.5.1 Etapas del método de Jenkins-Traub

El método se divide en tres etapas de la siguiente manera:

Etapa uno

Se inicializa $H^{0}(z)=P'(z)$. Con $S_0=0$ se calcula una nueva $H^{k}(z)$ con la siguiente fórmula,

$$H^{k+1}(j)=P(j)-H^{k+1}(j-1)\frac{H^{k}(0)}{P(0)},\text{ con } H^{k+1}(1)=P(1).$$

Se itera para acentuar las raíces de menor módulo. En la práctica es imposible indicar cuántas iteraciones son las adecuadas, pero se recomienda que sean al menos cinco.

Etapa dos

Se propone una raíz de la forma $S_k=\beta\cdot e^{i\theta}$. Con esta aproximación inicial, se calcula una nueva aproximación con

$$S_{k+1}=S_{k}-\frac{P(s)}{H^{k}(s)}$$

Se calcula una Q^{k+1} con la fórmula

$$Q^{k+1}(j)=Q^{k+1}(j-1)\cdot S_{k+1}+P(j),\text{ para } j=2,\ldots,n;\text{ con } Q^{k+1}(1)=P(1)$$

Se calcula una nueva H^{k+1} utilizando la fórmula (3.74)

$$H^{k+1}(z)=\sum_{j=2}^{n}\left[Q^{k+1}(j)-\frac{P(s)}{H^{k}(s)}\cdot Q^{k+1}(j-1)\right]$$

www.FreeLibros.me

Etapa tres

En esta etapa simplemente se repite el proceso de la etapa dos, sin modificar la raíz; es decir $S_{k+1} = S_k$. Una vez que se hacen varias iteraciones, se regresa a la etapa dos para aproximar más la raíz. La razón de estas iteraciones es, por tanto, simplemente acentuar la raíz de menor módulo.

En la sección 3.7.9 se desarrolla el código del método en programación Matlab. El método se aplica a un polinomio con coeficientes complejos, y al final se obtiene la raíz. Se aplica el proceso de deflación para encontrar las raíces del polinomio.

EJEMPLO 3.12

En el circuito de la figura 3.2, el interruptor K se cierra en tiempo igual a cero. Calcular el tiempo de estabilización del voltaje del capacitor, tomando como referencia que se estabiliza cuando llega al 99% de su voltaje final.

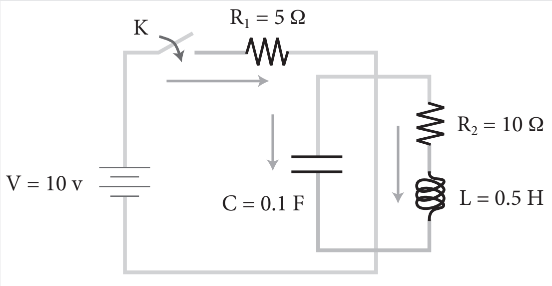


Figura 3.2 Circuito eléctrico.

SOLUCIÓN. La ecuación que modela el comportamiento físico del voltaje en el capacitor es la siguiente:

$$v_C = L\frac{di_L}{dt} + R_2 i_L$$

donde, de acuerdo con la conectividad, la corriente i_L que pasa por el inductor es:

$$\frac{d^2 i_L}{dt^2} + \left(\frac{1}{CR_1} + \frac{R_2}{L}\right)\frac{di_L}{dt} + \left(\frac{R_1 + R_2}{LCR_1}\right) i_L = \frac{V}{LCR_1}$$

Sustituyendo valores, se tiene

$$\frac{d^2 i_L}{dt^2} + 22\frac{di_L}{dt} + 60 i_L = 40$$

La solución homogénea de esta ecuación se establece con el polinomio $r^2 + 22r + 60 = 0$, el cual tiene como raíces: $r_1 = -3.189750$ y $r_2 = -18.810249$. Así, la corriente que pasa por el inductor es

$$i_L = -0.802801 e^{-3.189750t} + 0.136135 e^{-18.810249t} + \frac{2}{3}$$

Por tanto el voltaje en el capacitor es:

$$v_C = 0.5\frac{di_L}{dt} + 10 i_L$$

Sustituyendo valores se tiene

$$v_C = -6.747650 e^{-3.189750t} + 0.080983 e^{-18.810249t} + \frac{20}{3}$$

www.FreeLibros.me

La raíz más pequeña es la que tardará más en atenuarse; por tanto, es la que indicará cuándo llega el capacitor a su punto de estabilidad; así, cuando el término $-6.747650e^{-3.189750t} = -0.067476$, se llega al punto de estabilidad. Por tanto, el tiempo es: $t = 1.443740$ segundos. La figura 3.3 muestra la forma gráfica del voltaje del capacitor. Ahí se puede observar cómo se estabiliza en el tiempo calculado.

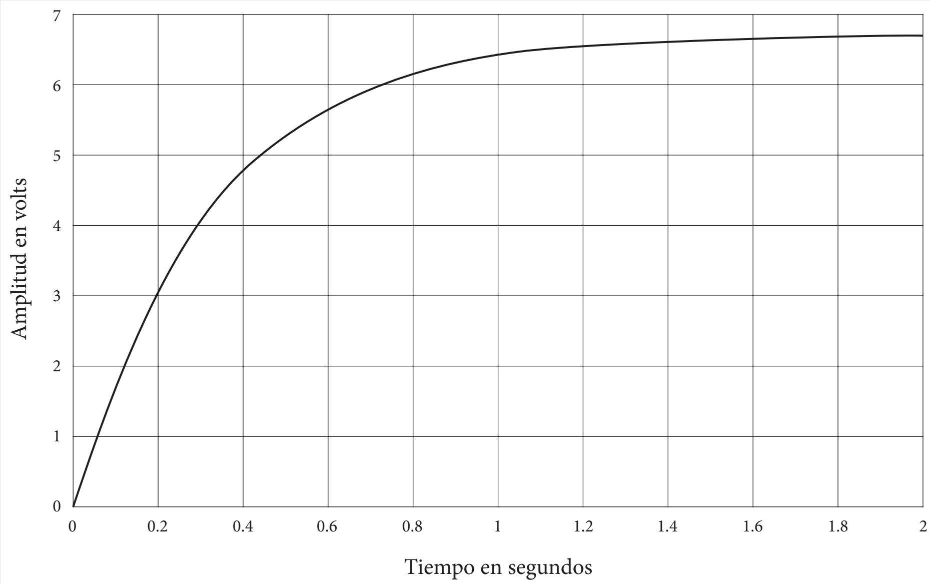


Figura 3.3 Gráfica del voltaje del capacitor.

3.6 Comparación de métodos

La aproximación más sencilla para los que escriben su propio código es el uso del algoritmo de la diferencia de cocientes para obtener la aproximación de las raíces, y un método de buena convergencia para mejorar las raíces, como es el método de Newton para raíces simples, o el método de Bairstow para raíces complejas conjugadas. Si se tiene a la mano un programa que use el método de Lehmer-Schur, éste se puede usar en vez del algoritmo de diferencia de cocientes. La solución completa de un polinomio no es trivial, y si se tiene un programa confiable para encontrarlas, se debe usar. Sin embargo, quedarán siempre polinomios que no se pueden resolver numéricamente por medios convencionales, y requieren análisis matemático cuidadoso para obtener la respuesta significativa.

3.7 Programas desarrollados en Matlab

En esta sección se concentran los códigos de los programas desarrollados para el capítulo. Enumerándolos se tienen los siguientes:

3.7.1 División sintética por un factor simple
3.7.2 División sintética por un factor cuadrático
3.7.3 Método de Bairstow
3.7.4 Método de Laguerre
3.7.5 Método de Bernoulli
3.7.6 Método de Newton
3.7.7 Algoritmo de diferencia de cocientes
3.7.8 Método de raíz cuadrada de Graeffe
3.7.9 Método de Jenkins-Traub

www.FreeLibros.me

3.7.1 División sintética por un factor simple

La división sintética por un factor simple es una forma compacta de realizar una división de un polinomio de grado *n* entre un polinomio de primer grado. Como resultado se tiene un cociente y un residuo. Cuando se utiliza como factor del polinomio una de sus raíces, el residuo será cero; de otra forma tendrá un valor diferente de cero. En Matlab, cuando se usa el signo % significa que se está haciendo un comentario. Adicionalmente, las funciones propias de Matlab aparecen sombreadas.

Programa principal de la división sintética por un factor simple

```
%  División sintética de un polinomio con la estructura P(x)= k1*x^n + k2*x^(n-1) +
%  ... + km*x + kn, donde se tiene que k1=1 por un factor simple f(x) = x + r.
%
%  El método entrega el cociente y el residuo después de aplicar el procedimiento de
%  deflación.
%
clear all
clc
%  Coeficientes del polinomio.
F = [1 56 1288 15680 108304 420224 836352 645120];
% Número de coeficientes del polinomio
n = length(F);
%  Factor simple de la forma x + r = 0. Para hacer la división se les cambia
%  directamente el signo r. Así se tiene que
f = [1    12];  % Coeficientes del factor simple.
r = -f(2);      % r utilizada en el proceso de deflación.
%  Inicia los polinomios para calcular la división por el factor simple.
P1(1:n) = zeros;
%  Hace la deflación del factor simple; es decir, es la división sintética de P(x).
P1(1) = F(1);
for k=2:n;
    P1(k) = F(k) + P1(k-1)*r;
end
%  Cociente resultante de la división sintética.
Q = P1(1:n-1)
%  Residuo resultante de la división sintética.
Rs = P1(n)
```

NOTA: Si se ejecuta este programa tal y como está, da como resultado final:

```
Q =          1         44         760         6560        29584       65216       53760
Rs =     0
```

Lo cual corresponde al cociente y al residuo de la división sintética.

3.7.2 División sintética por un factor cuadrático

La división sintética por un factor cuadrático es una forma compacta de realizar una división de un polinomio de grado *n*, donde *n* es al menos 3, entre un polinomio de grado 2. Como resultado se tiene un cociente y un residuo. Cuando se utiliza con un factor del polinomio, el residuo será cero. De otra forma tendrá un valor diferente de cero. En Matlab, cuando se usa el signo % significa que se está haciendo un comentario. Adicionalmente, las funciones propias de Matlab aparecen sombreadas.

Programa principal de la división sintética por un factor cuadrático

```
%  División sintética de un polinomio con la estructura
%  P(x)= k1*x^n + k2*x^(n-1) + ... + km*x + kn, donde se tiene que k1 = 1 por un
%  factor cuadrático f(x) = x^2 + p*x + q.
%  El método entrega el cociente y el residuo después de aplicar el procedimiento de
%  deflación.
```

www.FreeLibros.me

```
clear all
clc
%  Coeficientes del polinomio
P = [1   100   4335   106800   1646778   16486680   107494190   444647600
       1094071221   1396704420   654729075];
%  Número de coeficientes del polinomio.
N = length(P);
%  Factor cuadrático de la forma x^2 + p*x + q = 0. Para hacer la división se les
%  cambia directamente el signo a p y a q,; así se tiene que
f = [1    22    85];   % Coeficientes del factor cuadrático.
p = -f(2);             % p utilizada en el proceso de deflación.
q = -f(3);             % q utilizada en el proceso de deflación.
%  Inicia los polinomios para calcular la división por el factor cuadrático.
P1(1:N)   = zeros;
%  División sintética de P(x).
P1(1) = P(1);
P1(2) = P(2) + P1(1)*p;
for k=3:N;
    P1(k) = P(k) + P1(k-1)*p + P1(k-2)*q;
end
%  Cociente resultante de la división sintética.
Q = P1(1:N-2)
%  Residuo resultante de la división sintética.
Rs = P1(N-1:N)
```

NOTA: Si se ejecuta este programa tal y como está, da como resultado final:

```
Q  =   1   78   2534   44422   454104   2720522   9043866   14438178   7702695
Rs =    0    0
```

Lo cual corresponde al cociente y al residuo de la división sintética.

3.7.3 Método de Bairstow

El método de Bairstow se programó en forma dividida. En un programa se hace la función como tal, y en otro se hace la lógica para calcular los factores cuadráticos que contenga un polinomio de grado *n*. En Matlab, cuando se usa el signo % significa que se está haciendo un comentario. Adicionalmente, las funciones propias de Matlab aparecen sombreadas.

Programa principal del método de Bairstow codificado en Matlab

```
%  Programa principal que llama al método de Bairstow para cálculo de factores
%  cuadráticos.
clear all
clc
%  Vector de coeficientes.
Cf = [1 37 520 3490 11449 16633 8190];
n  = length(Cf);    % Longitud del vector de coeficientes.
po = 10;            % Inicia el factor q.
qo = 40;            % Inicia el factor p.
Dat = [];           % Inicia la matriz para almacenar los factores resultantes.
%  Inicia ciclo para calcular los factores cuadráticos de un polinomio de orden n,
%  para lo cual se deben tener más de tres coeficientes.
while n > 3
    % Ejecuta la función del método de Bairstow.
    [p,q,Pn]=Bairstow(Cf,n,po,qo);
    % Número de iteraciones que realizó el método hasta la convergencia.
    Nf = length(p);
    % Almacena los factores calculados de la forma Dat = [1  p  q],
    % lo que indica que se tiene un polinomio f(x) = x^2 + p*x + q.
    Dat = [Dat
           1   p(Nf)   q(Nf)];
```

www.FreeLibros.me

```
    % Asigna el residuo a un nuevo polinomio para sacar un nuevo factor cuadrático.
    Cf = Pn;
    % Saca la longitud del residuo resultante.
    n  = length(Cf);
    % Inicia nuevamente p y q con cualquier valor.
    po = 5;
    qo = 10;
end
%  Condicional para almacenar el último residuo. Si resulta que el residuo tiene tres
%  coeficientes, corresponde a un factor cuadrático, y si tiene sólo dos coeficientes,
%  corresponde a un factor simple.
if n == 3
    Dat = [Dat
           1   Pn(2)   Pn(3)]
elseif n == 2
    Dat = [Dat
           0   Pn(1)   Pn(2)]
end
```

Función llamada Bairstow codificada en Matlab

```
%  Función del método de Bairstow para el cálculo de factores cuadráticos de
%  polinomios con la estructura P(x)= k1*x^n + k2*x^(n-1) + ... + km*x + kn donde se
%  tiene que k1=1.
%
%  El método calcula un factor cuadrático de la forma:
%
%              f(x) = x^2 + p*x + q
%
%  La función se llama de la siguiente manera:
%
%              [p,q,Pz]=Bairstow(Cf1,N,po,qo)
%
%  Entradas:
%              Cf1 -- Vector de coeficientes.
%              N   -- Número de coeficientes.
%              po  -- Inicia el factor p.
%              qo  -- Inicia el factor q.
%
%  Salidas:
%              p   -- Factor p final.
%              q   -- Factor q final.
%              Pz  -- Factores del residuo resultante.
%
function [p,q,Pz]=Bairstow(Cf1,N,po,qo)
%  Valores iniciales de p y q
p(1) = po;
q(1) = qo;
%  Inicia los polinomios para calcular las divisiones por factores cuadráticos de los
%  polinomios P(x), x*Q(x) y Q(x).
P1(1:N)   = zeros;
P2(1:N-1) = zeros;
P3(1:N-2) = zeros;
%  Incrementos iniciales para el ciclo iterativo.
Dp = 1;
Dq = 1;
tol = 1e-6;        % Tolerancia de convergencia.
ct = 1;            % Contador de iteraciones.
%  Inicia el ciclo iterativo.
while abs(Dp) > tol | abs(Dq) > tol
    % División sintética de P(x).
    P1(1) = Cf1(1);
    P1(2) = Cf1(2) - P1(1)*p(ct);
    for k=3:N;
        P1(k) = Cf1(k) - P1(k-1)*p(ct) - P1(k-2)*q(ct);
    end
```

www.FreeLibros.me

```
    % División sintética de x*Q(x).
    Cf2 = [P1(1:N-2) 0];
    P2(1) = Cf2(1);
    P2(2) = Cf2(2) - P2(1)*p(ct);
    for k=3:N-1;
        P2(k) = Cf2(k) - P2(k-1)*p(ct) - P2(k-2)*q(ct);
    end
    % División sintética de Q(x) cuyo resultado son las primeras n-1 operaciones de
    % la división sintética de x*Q(x).
    P3 = P2(1:N-2);
    % Asignación de los residuos para la formación de las matrices A y B.
    R1 = P1(N-1);     R2 = P2(N-2);     R3 = P3(N-3);
    S1 = P1(N);       S2 = P2(N-1);     S3 = P3(N-2);
    A = [-R2 -R3
         -S2 -S3];
    B = [-R1
         -S1];
    % Cálculo de los incrementos a p y q.
    Dpq = inv(A)*B;
    Dp  = Dpq(1);
    Dq  = Dpq(2);
    % Asignación de los incrementos y almacenamiento.
    ct    = ct + 1;
    p(ct) = p(ct-1) + Dp;
    q(ct) = q(ct-1) + Dq;
end
%  Polinomio resultante después de hacer la deflación por un factor cuadrático.
Pz = P1(1:N-2);
```

NOTA: Si se ejecuta este programa tal y como está, da como resultado final:

```
Dat =
         1.0000    12.0000     35.0000
         1.0000     3.0000      2.0000
         1.0000    22.0000    117.0000
```

Lo cual corresponde a los tres factores cuadráticos del polinomio de orden 6, cuyos coeficientes se dan en forma inicial en el arreglo Cf. Se debe mencionar que el programa es general, basca cambiar los coeficientes y que el primero sea uno, y entregalos factores cuadráticos y uno simple si fuera el caso de un polinomio de grado impar.

3.7.4 Método de Laguerre

El método de Laguerre se programó en dos partes, en un programa se hace la función como tal, y en otro se hace la lógica para calcular las raíces que contenga un polinomio de grado n. Se utiliza la división sintética para pasar de un polinomio de grado n a uno de grado $n-1$, después de encontrar una raíz. En Matlab, cuando se usa el signo % significa que se está haciendo un comentario, adicionalmente, las funciones propias de Matlab aparecen sombreadas.

Programa principal en Código Matlab del método de Laguerre

```
%  Programa principal que llama al método de Laguerre para cálculo de raíces reales
%  simples
clear all
clc
%  Vector de coeficientes del polinomio,
F = [1    37.06    565.9508    4596.67151    21297.81011979
     55616.3956104624    74044.983332884    36780.5723406719];
N  = length(F);         % Número de coeficientes del polinomio.
n  = length(F)-1;       % Grado del polinomio.
Xo = 77;                % Inicia con un valor arbitrario.
```

www.FreeLibros.me

```
Raices = [];            % Inicia la matriz para almacenar las raíces resultantes.
%  Inicia ciclo para calcular las raíces de un polinomio de orden n.
for k = 1:N-2
    % Ejecuta el método de Laguerre.
    [R]=Laguerre(F,n,Xo);
    % Número de iteraciones que realizó el método hasta la convergencia.
    Nf = length(R);
    % Valor de la raíz calculada.
    Ra = round(R(Nf)*1e6)/1e6;
    % Almacena los factores calculados de la forma Raíces = [raíz].
    Raices = [Raices
              Ra];
    % Hace la deflación del factor simple; es decir, es la división sintética de P(x).
    P1(1) = F(1);
    for k=2:n;
        P1(k) = F(k) + P1(k-1)*Ra;
    end
    % Asigna el residuo a un nuevo polinomio.
    F = P1;
    % Obtiene el grado del  residuo resultante.
    n  = length(F)-1;
    % Inicia nuevamente X con cualquier valor.
    X(1) = 77;
    clear P1
end

% Valor de la última raíz
Ra = round(-F(2)*1e6)/1e6;
% Almacena la última raíz.
Raices = [Raices
          Ra];
```

Función llamada Laguerre codificada en Matlab

```
%  Función del método de Laguerre para calcular las raíces reales simples de
%  polinomios con la estructura P(x)= k1*x^n + k2*x^(n-1) + ... + km*x + kn donde se
%  tiene que k1=1.
%
%  El método calcula la raíz más cercana a la condición dada.
%
%  La función se llama de la siguiente manera:
%
%              [X]=Laguerre(F,n,Xo)
%
%  Entradas:
%              F  -- Vector de coeficientes.
%              n  -- Grado del polinomio.
%              Xo -- Condición inicial.
%
%  Salida:
%              X  -- Vector que contiene todas las aproximaciones.
%                    hasta la convergencia.
function [X]=Laguerre(F,n,Xo);
%  Inicia el vector de salida con la condición inicial.
X(1) = Xo;
%  Cálculo de la primera derivada de F.
Fp = polyder(F);
%  Cálculo de la segunda derivada de F.
Fpp = polyder(Fp);
Dx   = 20;          % Inicial valor para entrar en el while.
tol = 1e-6;         % Fija la tolerancia de convergencia.
ct  = 1;            % Contador de iteraciones hasta la convergencia.
%  Inicia el ciclo iterativo.
while abs(Dx) > tol
    % Cálculo del polinomio de Laguerre.
```

www.FreeLibros.me

```
    x  = X(ct);
    f0 = polyval(F,x);
    f1 = polyval(Fp,x);
    f2 = polyval(Fpp,x);
    H  = (n-1)^2 * f1^2 -n*(n-1)*f0*f2;
    % Cuenta la siguiente iteracion.
    ct = ct+1;
    % Asigna el signo de la derivada.
    as = sign(f1);
    % Calcula la siguiente iteración.
    X(ct) = X(ct-1) - (n*f0)/(f1+as*sqrt(H));
    % Calcula la diferencia entre dos iteraciones consecutivas para verificar la
    %  convergencia.
    Dx = abs(X(ct)-X(ct-1));
end
```

NOTA: Si se ejecuta este programa tal y como está, da como resultado final:

```
Raíces =
            -1.12      -3.44      -4.56      -4.99      -5.67      -7.83      -9.45
```

Lo cual corresponde a las siete raíces del polinomio de orden 7, cuyos coeficientes se dan en forma inicial en el arreglo F. Se debe mencionar que el programa es general. Basta cambiar el arreglo de coeficientes y que el primero sea uno, y muestra todas las raíces.

3.7.5 Método de Bernoulli

El método de Bernoulli se programó en dos secciones. En uno se hace la función como tal, y en otro se hace la lógica para calcular las raíces que contenga un polinomio de grado n. Se utiliza la división sintética para pasar de un polinomio de grado n a uno de grado $n-1$, después de encontrar una raíz. En Matlab, cuando se usa el signo % significa que se está haciendo un comentario.

Programa principal en código Matlab del método de Bernoulli

```
%  Programa principal que llama al método de Bernoulli para calcular las raíces
%  reales simples.
clear all
clc
%  Vector de coeficientes del polinomio.
F = [1   155   10770   441750   11844273   216903435   2747429180   23767101700
        134376696576   448372820160   670442572800];
N        = length(F);       % Número de coeficientes del polinomio.
n        = length(F)-1;     % Grado del polinomio.
Raices = [];                % Inicia la matriz para almacenar las raíces resultantes.
%  Inicia ciclo para calcularlas raíces de un polinomio de orden n.
for k = 1:N-2
    % Ejecuta el método de Bernoulli.
    [R]=Bernoulli(F,n);
    % Número de iteraciones que realizó el método hasta la convergencia.
    Nf = length(R);
    % Valor de la raíz calculada.
    Ra = round(R(Nf)*1e4)/1e4;
    % Almacena los factores calculados de la forma Raíces = [raíz].
    Raices = [Raices
                  Ra];
    % Hace la deflexión del factor simple; es decir, es la división sintética de P(x).
    P1(1) = F(1);
    for k=2:n;
        P1(k) = F(k) + P1(k-1)*Ra;
    end
    % Asigna el residuo a un nuevo polinomio.
```

www.FreeLibros.me

```
    F = P1;
    % Saca el grado del residuo resultante.
    n  = length(F)-1;
    clear P1
end
% Valor de la ultima raíz.
Ra = round(-F(2)*1e4)/1e4;
% Almacena la ultima raíz
Raices = [Raices
          Ra]
```

Función llamada Bernoulli codificada en Matlab

```
%  Función del método de Bernoulli para calcular las raíces reales simples de
%  polinomios con la estructura P(x)= k1*x^n + k2*x^(n-1) + ... + km*x + kn donde se
%  tiene que k1=1.
%
%  El método calcula la raíz de mayor módulo.
%
%  La función se llama de la siguiente manera:
%
%              [R]=Laguerre(F,n)
%
%  Entradas:
%              F  -- Vector de coeficientes.
%              n  -- Grado del polinomio.
%
%  Salida:
%              R  -- Vector que contiene todas las aproximaciones hasta la
%                    convergencia.
%
function [Raiz]=Bernoulli(F,n);
A = zeros(n,n);
B = zeros(n,1);
% Ciclo iterativo para sacar las condiciones iniciales.
for k = 1:n
    a = 0;
    for l = k:-1:1
        a = a+1;
        A(k,a) = F(l);
    end
    B(k,1) = -k*F(k+1);
end
Coef = inv(A)*B;
% Ciclo para calcular una raíz hasta la convergencia.
Raiz(1) = 1;
tol = 1;
k = length(F);
while tol > 1e-6
    Coef(k) = sum(-F(2:n+1)'.*Coef(k-1:-1:k-n));
    Raiz(k-n+1) = Coef(k)/Coef(k-1);
    k = k + 1;
    tol = abs(Raiz(k-n)-Raiz(k-n-1));
end
```

NOTA: Si se ejecuta este programa tal y como está, da como resultado final:

```
Raíces =
             -20          -19          -18          -17          -16
             -15          -14          -13          -12          -11
```

Lo cual corresponde a las diez raíces del polinomio cuyos coeficientes se dan en forma inicial en el arreglo F. Se debe mencionar que el programa es general y que entrega las raíces de mayor a menor módulo, como se verifica en el ejercicio de Matlab.

www.FreeLibros.me

3.7.6 Método de Newton

El método de Newton encuentra la raíz de menor módulo, independientemente de si es compleja o real. Después de encontrar una raíz, se utiliza la división sintética para pasar de un polinomio de grado n a uno de grado $n-1$. El programa está organizado en dos secciones: en un programa se hace la función como tal, y en otro se hace la lógica para calcular las raíces que contenga un polinomio de grado n. En Matlab, cuando se usa el signo % significa que se está haciendo un comentario.

Programa principal en código Matlab del método de Newton

```
%  Programa principal que llama al método de Newton para calcular las raíces simples.
clear all
clc
% Vector de coeficientes del polinomio.
F = [ 1  26  270   1420  29   5274    2520];
N  = length(F);          % Número de coeficientes del polinomio.
n  = length(F)-1;        % Grado del polinomio.
Raices = [];             % Inicia la matriz para almacenar las raíces resultantes.
%  Inicia el ciclo para calcular las raíces de un polinomio de orden n.
for k = 1:N-2
    % Ejecuta el método de Newton.
    [R,DR]= Newton(F,n);
    % Número de iteraciones que realizó el método hasta la convergencia.
    Nf = length(R);
    % Valor de la raíz calculada.
    Ra = R(Nf);
    % Almacena los factores calculados de la forma Raíces = [raíz]
    Raices = [Raices
              Ra];
    % Hace la deflexión del factor simple; es decir, es la división sintética de P(x).
    P1(1) = F(1);
    for k=2:n;
        P1(k) = F(k) + P1(k-1)*Ra;
    end
    F = P1;              % Asigna el residuo a un nuevo polinomio.
    n  = length(F)-1;    % Saca el grado del residuo resultante.
    clear P1
end
%  Valor de la última raíz.
Ra = -F(2);
%  Almacena la última raíz.
Raices = [Raices
          Ra]
```

Función llamada Newton codificada en Matlab

```
%  Función del método de Newton para calcular las raíces simples de polinomios con la
%  estructura P(x)= k1*x^n + k2*x^(n-1) + ... + km*x + kn donde se tiene que k1=1.
%
%  El método calcula la raíz de menor módulo.
%
%  La función se llama de la siguiente manera:
%
%               [Z,dzk]= Newton(F,n)
%
%  Entradas:
%               F  -- Vector de coeficientes.
%               n  -- Grado del polinomio.
%
%  Salida:
%               Z   -- Vector que contiene todas las aproximaciones hasta la
%                      convergencia.
```

www.FreeLibros.me

```
%               dzk -- Vector que contiene todos los incrementos calculados.
%
function [Z,dzk]=Newton(F,n)
Fp = polyder(F);                 % Cálculo de la primera derivada de F.
Z(1) = 0 ;                       % Raíz para fijar la condición de arranque.
f0   = polyval(F,Z(1));          % Evaluación del polinomio en la raíz.
fp   = polyval(Fp,Z(1));         % Evaluación de la derivada en la raíz.

%  Cálculo del primer incremento dzk.
if fp ~= 0
    dzk(1) = -f0/fp;
    else
    dzk(1) = 1;
end
%  Ciclo iterativo para calcular Z2.
for k = 1:n
    al(k) = (abs(F(n+1)/F(k)))^(n+1-k)  *  (dzk(1)/abs(dzk(1)));
end
[x,y] = min(abs(al));   % Regresa el mínimo módulo y su posición.
pn    = sign(al(y));    % Toma el signo del mínimo módulo.
Z(2)  = (1/2)*pn*x;     % Calcula la primera aproximación a la raíz.
%  Primera etapa que concluye cuando se cumple dos veces consecutivas con el criterio
%  de convergencia.
tol = 1;    % Inicia la tolerancia de convergencia.
kr  = 3;    % Apuntador de iteraciones.
while tol <= 3
    f0 = polyval(F,Z(kr-1));      % Evaluación del polinomio en la raíz.
    fp = polyval(Fp,Z(kr-1));     % Evaluación de la derivada en la raíz.
    wk = -f0/fp;                  % Se calcula el radio de aproximación.
    % Condicional para elegir dzk.
    if abs(wk) <= 3*abs(Z(kr-1)-Z(kr-2))
       dzk(kr-1) = wk;
    else
        dzk(kr-1) = (3*abs(Z(kr-1)-Z(kr-2))*wk*exp(i*pi/4))/abs(wk);
    end
    Z(kr) = Z(kr-1)+dzk(kr-1);    % Calcula la nueva aproximación de la raíz.
    f1    = polyval(F,Z(kr));     % Evaluación del polinomio en la raíz.
    % Condicional de la evaluación del polinomio con aproximaciones consecutivas,
    % para buscar nuevas aproximaciones.
    if abs(f1) > abs(f0)
        ra    = Z(kr-1)+(2:n)*dzk(kr-1);    % Raíces nuevas con m = 2,..., n.
        faux  = polyval(F,ra);              % Evaluación de la función con las raíces.
        [x,y] = min(abs(faux));             % Regresa el mínimo módulo y su posición.
        Zaux  = ra(y);                      % Elección del menor módulo.
    else
        ra    = Z(kr-1)+[1/2 1/4 (1/4)*exp(i*pi/4)]*dzk(kr-1);   % Raíces nuevas con
                                                                 % m = 1/2 ...
        faux  = polyval(F,ra);                                   % Evaluación de la
                                                                 % función con las
                                                                 % raíces.
        [x,y] = min(abs(faux));                                  % Regresa el mínimo
                                                                 % módulo y su
                                                                 % posición.
        Zaux  = ra(y);                                           % Elección del menor
                                                                 % módulo.
    end
    % Condicional para cambiar dirección en caso de no encontrar un paso adecuado.
    if abs(Z(kr)) > abs(Zaux)
        dzk(kr) = -(1/2)*exp(i*pi/4)*dzk(kr-1);
        Z(kr)   = Z(kr-1)+dzk(kr);
    else
        dzk(kr) = Zaux - Z(kr-1);
        Z(kr) = Zaux;
    end
    % Evaluación de la prueba de convergencia.
    fkd  = polyval(F,Z(kr));
```

www.FreeLibros.me

```
    fkp1 = polyval(Fp,Z(kr-1));
    fkp2 = polyval(Fp,Z(kr));
    CDd  = 2*abs(fkd)*abs(fkp1-fkp2);
    Cdi  = (abs(fkp2))^2*abs(Z(kr-1)-Z(kr));
    if CDd <= Cdi
        tol = tol+1;
    else
        tol = tol-1;
    end
    kr = kr+1;
end
%  Si cumplió dos veces con el criterio de convergencia, se pasa a la etapa 2. En
%  esta etapa se evalúa directamente la regla de Newton con la formula Z(n)=Z(n-
%  1)+dzk(n).
while tol > 1e-12                             % Error absoluto permitido.
    f0 = polyval(F,Z(kr-1));                  % Evaluación del polinomio en la raíz.
    fp = polyval(Fp,Z(kr-1));                 % Evaluación de la derivada en la raíz.
    dzk(kr) = -f0/fp;                         % Cálculo del incremento.
    Z(kr)   = Z(kr-1)+dzk(kr);                % Cálculo de la raíz.
    tol = abs(abs(Z(kr)) - abs(Z(kr-1)));     % Error absoluto.
    kr = kr+1;
end

%  Condicional para quitar la parte imaginaria cuando es más pequeña que 1e-6.
a=real(Z(kr-1));
b=abs(imag(Z(kr-1)));
if b < 1e-6
    Z(kr-1)=a;
end
```

NOTA: Si se ejecuta este programa tal y como está, da como resultado final:

```
Raices =
             -0.4556
              0.5325 + 1.8410i
              0.5325 - 1.8410i
            -14.1176
             -6.2459 - 8.2257i
             -6.2459 + 8.2257i
```

Lo cual corresponde a las seis raíces del polinomio cuyos coeficientes se dan en forma inicial en el arreglo F. Se debe mencionar que el programa es general y que entrega las raíces de menor a mayor módulo, como se verifica en el ejercicio de Matlab.

3.7.7 Algoritmo de diferencia de cocientes

El algoritmo de diferencia de cocientes encuentra todas las raíces de un polinomio de grado n en forma simultánea, es decir, hace la aproximación de las raíces y las va precisando paso a paso, todas al mismo tiempo. Al final las organiza de manera que quedan de mayor a menor módulo. El programa está organizado de manera sencilla en dos secciones; una de ellas es el código del algoritmo y la otra es el código principal que llama a esta función. En Matlab, cuando se usa el signo % significa que se está haciendo un comentario.

Programa principal en código Matlab del método de diferencia de cocientes

```
%  Método de Diferencia de Cocientes para cálculo de todas las raíces de un polinomio
%  de grado n.
clear all
clc
%  Coeficientes del polinomio.
F = [1    41.67    734.0079    7120.893153    41505.25837644    148422.750838295
317083.244148468       368985.006030525       178708.14045363];
```

www.FreeLibros.me

```
% Número de coeficientes del polinomio.
n = length(F);
%  Ejecuta el método de diferencia de cocientes.
[Q,E,R]=DiferenciaDeCocientes(F,n);
%  Valor de las raíces de mayor a menor módulo
R
```

Función llamada Diferencia DeCocientes codificada en Matlab

```
%  Función del método de diferencia de cocientes para cálculo de las raíces de
%  polinomios con la estructura P(x)= k1*x^n + k2*x^(n-1) + ... + km*x + kn donde se
%  tiene que k1=1.
%
%  La función se llama de la siguiente manera:
%
%              [Q,E,R]=DiferenciaDeCocientes(F,n).
%
%  Entradas:
%              F  -- Vector de coeficientes.
%              n  -- Grado del polinomio.
%  Salida:
%              Q  -- Matriz que contiene todos los renglones de las aproximaciones
%                    hasta la convergencia.
%              E  -- Matriz que contiene todos los errores de las aproximaciones
%                    hasta la convergencia.
%              R  -- Vector que contiene las raíces calculadas.
%
function[Q,E,R]=DiferenciaDeCocientes(F,n);
%  Inicializa Q.
Q(1,1) = - F(2)/F(1);
Q(1,2:n-1) = 0;
%  Inicializa E.
E(1,1:n) = 0;
E(1,2:n-1) = F(3:n)./F(2:n-1);
m  = 1;            % Inicia el contador de iteraciones.
dE = 1;            % Inicia la tolerancia para entrar en el ciclo while.
tol = 1e-6;        % Tolerancia para convergencia.
%  Inicia el ciclo iterativo.
while dE > tol
   m = m +1;
   % Calcula el renglón de Q.
   Q(m,1:n-1) = Q(m-1,1:n-1) + E(m-1,2:n) - E(m-1,1:n-1);
   %  Calcula el renglón de E.
   E(m,2:n-1) = E(m-1,2:n-1) .* Q(m,2:n-1) ./ Q(m,1:n-2);
   %  Valor máximo del último renglón e.
   dE = max(abs(E(m,2:n-1)));
end
%  Valor final calculado de las raíces.
Ra = Q(m,:);
%  Saca la raíz redondeada a cuatro cifras decimales.
R = round(Ra*1e4)/1e4;
```

NOTA: Si se ejecuta este programa tal y como está, da como resultado final:

```
R =           -9.25               -8.21               -6.77               -5.32
       -4.56               -3.44               -2.34               -1.78
```

Lo cual corresponde a las ocho raíces del polinomio cuyos coeficientes se dan en forma inicial en el arreglo F. Se debe mencionar que el programa es general y que entrega las raíces en orden de mayor a menor módulo, como se verifica en el ejercicio de Matlab.

3.7.8 Método de raíz cuadrada de Graeffe

El método de raíz cuadrada de Graeffe encuentra los módulos de las raíces de un polinomio de grado n en forma simultánea. Al final las organiza de manera que quedan de mayor a menor módulo pero sin

www.FreeLibros.me

signo, de manera que se necesita utilizar el proceso de deflación para sacar el signo de cada raíz. El programa contiene dos rutinas; en una está el método como función y la otra contiene la forma general de ejecución del método.

Programa principal en código Matlab del método de Graeffe

```
%  Método de raíz cuadrada de Graeffe para el cálculo de las raíces de un polinomio
%  de grado n.
clear all
clc
%  Coeficientes del polinomio.
Coef = [1  45  805  7155  31594  55440];
%  Número de coeficientes del polinomio.
n = length(Coef);
%  Ejecuta el método de la raíz cuadrada de Graeffe.
[F,R]=Graeffe(Coef,n);
%  Valor de las raíces de mayor a menor módulo.
R
```

Función llamada Graeffe codificada en Matlab

```
%  Función del método de raíz cuadrada de Graeffe para calcular las raíces de
%  polinomios con la estructura P(x)= k1*x^n + k2*x^(n-1) + ... + km*x + kn donde se
%  tiene que k1=1.
%
%  El método, como tal, funciona con base en la potenciación de valores; por esta
%  razón no se aplica a polinomios de alto orden.
%
%  La función se llama de la siguiente manera:
%
%              [F,R]=Graeffe(F,n)
%
%  Entradas:
%              Fu  -- Vector de coeficientes.
%              n   -- Grado del polinomio.
%
%  Salida:
%              F  -- Matriz que contiene todos los renglones de las aproximaciones
%                    hasta la convergencia.
%              R  -- Matriz que contiene todas las aproximaciones de las raíces hasta
%                    la convergencia.
%

function [F,R]=Graeffe(Fu,n);
N = n-1;           % Grado del polinomio.
F(1,:) = Fu ;      % Inicia el arreglo de coeficientes.
k = 2;             % Contador para almacenar todas las iteraciones.
tol = 1;           % Inicia la tolerancia para entrar en el el ciclo while.
%  Ciclo condicional para fijar la máxima convergencia y el valor para redondear; la
%  convergencia es diferente. Depende del grado, debido a que el método utiliza en
%  forma explícita multiplicaciones sucesivas entre coeficientes y no se puede
%  obtener la convergencia hasta donde se quiera, pues se llega a tener sobreflujo
%  numérico.
if N == 3
    conv = 1e-3;
    redo = 1e3;
elseif N == 4 | N == 5 | N == 6
    conv = 5e-2;
    redo = 1e2;
end
%  Inicia el ciclo iterativo para el cálculo de raíces.
while tol > conv
    % Inicia el ciclo iterativo para la evaluación de los nuevos coeficientes.
```

www.FreeLibros.me

```
    for m = N:-1:0
        a = min([N-m m]);
        Ar = 0;
        for km = 1:a
            Ar = Ar + (-1)^(km)*(F(k-1,m-km+1))*(F(k-1,m+km+1));
        end
        F(k,m+1) = (-1)^(n-m)*(F(k-1,m+1)^2 + 2*Ar);
    end
    % Ciclo para calcular las aproximaciones a las raíces.
    rs = 1/(2.^(k-1));
    R(1,k)= abs(F(k,2)).^rs;
    for kl = 2:n-1
            disc = 1;
        for kh = 1:kl-1
            disc = disc*R(kh,k);
        end
        R(kl,k)=(1/disc)*abs(F(k,kl+1)).^rs;
    end
    % Tolerancia entre dos iteraciones consecutivas para todas las raíces.
    TOLER = abs(R(:,k)-R(:,k-1));
    tol = max(TOLER);    % El valor máximo de todas las tolerancias.
    k = k+1;             % Contador de iteraciones.
end
%  Valor de las raíces calculadas.
R = round(R*redo)/redo;
```

NOTA: Si se ejecuta este programa tal y como está, da como resultado final:

```
R =
    0      20.37      13.93      11.82      11.16      11.02      11
    0      12.73      10.84      10.16      10         10         10
    0      8.89       8.9        8.93       8.97       9          9
    0      6.21       7.3        7.81       7.97       8          8
    0      3.87       5.65       6.62       6.94       7          7
```

Lo cual corresponde a las cinco raíces del polinomio cuyos coeficientes se dan en forma inicial en el arreglo Coef. Se debe mencionar que el programa es general y que presenta las raíces en orden de mayor a menor módulo, como se verifica en el ejercicio de Matlab.

3.7.9 Método de Jenkins-Traub

El método de Jenkins-Traub encuentra la raíz más pequeña de un polinomio que tiene coeficientes complejos. El programa contiene varias rutinas en forma de función y un programa principal en el cual se utilizan las funciones creadas. Este programa contiene la forma general de ejecución del método.

Programa principal en código Matlab del método de Jenkins-Traub

```
%  Método de Jenkins Traub para el cálculo de raíces de polinomios con coeficientes
%  complejos
clear all
clc
OPR = [-5  4  2 -1  5  -4  4  2 -4  8  9]; % Parte real de los coeficientes.
OPI = [-7 -5  9  7  5  -7  6 -9  6  2  5]; % Parte imaginaria de los coeficientes.
P = OPR+OPI*i;                             % Vector de coeficientes complejos.
n = length(P);                             % Número de coeficientes del polinomio.
Ra = [];                                   % Inicia el espacio para guardar las raíces.
%  Ciclo iterativo paracalcular las raíces del polinomio.
for k = 1:n-1
    % En la etapa uno sólo se modifica la matriz H tomando s=0.
    [H] = EtapaUno(n,P);
    % En la etapa dos se propone una raíz y se itera hasta la convergencia.
    S = 1e-3*exp((49/180)*pi*i);  % Se propone un valor de la raíz a un ángulo de 49
                                  % grados.
```

www.FreeLibros.me

```
    [S,H]=EtapaDos(n,P,S,H);
    % En la etapa tres se realizan 10 iteraciones sin cambiar la raíz para forzar a
    % que una sea dominante.
    [H]=EtapaTres(n,P,S,H);
    % En la etapa dos se propone una raíz y se itera hasta la convergencia.
    [Zr,H]=EtapaDos(n,P,S,H);
    % Hace la deflación del factor simple; es decir, es la división sintética de P(x).
    P1(1) = P(1);
    for k=2:n;
        P1(k) = P(k) + P1(k-1)*Zr;
    end
    Q = P1(1:n-1);    % Cociente resultante de la división sintética.
    R = P1(n);        % Residuo resultante de la división sintética.
    clear P1          % Se limpia el espacio donde se realiza la división sintética.
    P = Q;            % Se asigna el cociente al nuevo polinomio de grado n-1.
    n = length(P);    % Número de coeficientes del nuevo polinomio después de la
                      % deflación.
    % Guarda la raíz calculada.
    Ra = [Ra
          Zr];
end
Ra      % Las N raíces finales del polinomio de grado N
```

Función de Matlab llamada EtapaUno

```
function[H] = EtapaUno(n,P)
% Se determina la primera aproximación de H(z)=P'(z).
for K = 1 : n-1
    H(K) = (n-K)*P(K)/(n-1);
end
% Realizar n iteraciones para la primera etapa.
for km = 1:10
    % Dividir -H(0)/P(0), lo que equivale a dividir  -H(n-1)/P(n).
    Td = -H(n-1)/P(n);
    H(1) = P(1);
    for kl=1:n-2
        H(n-kl) = Td*H(n-kl-1) + P(n-kl);
    end
end
```

Función de Matlab llamada EtapaDos

```
function[S,H] = EtapaDos(n,P,S,H);
% Vector de S^n para evaluar los polinomios.
[Sp,Sh]=VectorS(S,n);
% Cálculo de -P(S)/H(s).
Td = -sum(P.*Sp)/sum(H.*Sh);
% Se actualiza la raíz S.
S = S + Td;
% Inician las iteraciones.
for k = 1:20
    % Calcular un nuevo polinomio H(s).
    [H]=NuevoH(n,Td,S,P);
    % Vector de S^n para evaluar los polinomios,
    [Sp,Sh]=VectorS(S,n);
    % Cálculo de -P(S)/H(s).
    Td = -sum(P.*Sp)/sum(H.*Sh);
    % Se actualiza la raíz S.
    S = S + Td;
end
```

Función de Matlab llamada EtapaTres

```
function [H]=EtapaTres(n,P,S,H);
% Vector de S^n para evaluar los polinomios.
```

www.FreeLibros.me

```
[Sp,Sh]=VectorS(S,n);
% Cálculo de -P(S)/H(s).
Td = -sum(P.*Sp)/sum(H.*Sh);
% Se realizan varias iteraciones sin cambiar la raíz para forzar a que la de menor
% módulo se separe de las demás.
for k = 1:10
    % Calcular un nuevo polinomio H(s).
    [H]=NuevoH(n,Td,S,P);
    % Cálculo de -P(S)/H(s).
    Td = -sum(P.*Sp)/sum(H.*Sh);
end
```

Función de Matlab llamada VectorS

```
function [Sp,Sh] = VectorS(S,n);
% Vector de S elevados a las diferentes potencia para evaluar el polinomio P(s).
Sp(n) = 1;
for k = 1:n-1
    Sp(k) = S.^(n-k);
end
% Vector de S elevados a las diferentes potencias para evaluar el polinomio H(s).
Sh = Sp(2:n);
```

Función de Matlab llamada NuevoH

```
function[H]=NuevoH(n,Td,S,P);
% Cálculo de la matriz auxiliar para la formación de la matriz H.
Q(1)=P(1);
Pv = P(1);
for k = 2:n-1
    Pv = Pv*S + P(k);
    Q(k) = Pv;
end
% Cálculo de la nueva H.
H(1)=P(1);
for k=2:n-1
    H(k) = Td*Q(k-1) + Q(k);
end
```

NOTA: Si se ejecuta este programa tal y como está, da como resultado final:

```
Ra =
        -0.496221106758525    -0.310855571248878i
        +0.924551080997739    +0.4554198598338i
        -0.332953946860917    +0.890584316949746i
        +0.297707724919078    +0.84049793871616i
        +0.116918277821752    -1.02137416304227i
        -0.649845629294501    -1.22189090269821i
        +0.795192136589549    -0.562084866761122i
        -1.36628834925816     -0.237760637045877i
        +1.28790445667054     -0.0800926948868546i
        -0.77966734752926     +0.531340503967289i
```

Lo cual corresponde a las n raíces del polinomio cuyos coeficientes se dan en forma inicial en el arreglo P. Se debe mencionar que el programa es general y que muestra las raíces, como se verifica en el ejercicio de Matlab.

www.FreeLibros.me

Problemas propuestos

3.8.1 Por el método de división sintética (por inspección) determine las raíces del siguiente polinomio:

$$f(x)=x^7-10x^6+14x^5+140x^4-511x^3+350x^2+496x-480$$

3.8.2 Por división sintética calcule el cociente del polinomio

$$P(x)=x^6-10x^5-60x^4+646x^3+491x^2-7\,116x+6\,048$$

si se sabe que $x=-7$ es una raíz del polinomio.

3.8.3 Por división sintética calcule el cociente del polinomio

$$P(z)=z^7-12z^6+42z^5+12z^4-375z^3+840z^2-748z+240$$

si se sabe que $z=5$ es una raíz del polinomio.

3.8.4 Por división sintética calcule el cociente del polinomio

$$P(y)=y^7-4y^6-62y^5+200y^4+1\,009y^3-2\,356y^2-3\,828y+5\,040$$

si se sabe que $y=1$ es una raíz del polinomio.

3.8.5 Por división sintética calcule el cociente del polinomio

$$P(a)=a^3+2a^2-a-2$$

si se sabe que $a=-2$ es una raíz del polinomio.

3.8.6 Por división sintética calcule el cociente del polinomio

$$P(d)=d^5-d^4-43d^3+61d^2+342d-360$$

si se sabe que $d=5$ es una raíz del polinomio.

3.8.7 Por división sintética calcule el cociente del polinomio

$$P(y)=y^5-2y^4-23y^3-112y^2-44y+1\,680$$

si se sabe que $P_1(y)=y^2+4y+20$ es un factor cuadrático del polinomio.

3.8.8 Por división sintética calcule el cociente del polinomio

$$P(y)=y^6-17y^5+80y^4-90y^3+249y^2-853y+630$$

si se sabe que $P_1(y)=y^2+2y+5$ es un factor cuadrático del polinomio.

3.8.9 Por división sintética calcule el cociente del polinomio

$$P(y)=y^6+19y^5+154y^4+664y^3+1\,543y^2+1\,765y+750$$

si se sabe que $P_1(y)=y^2+8y+25$ es un factor cuadrático del polinomio.

3.8.10 Por división sintética calcule el cociente del polinomio

$$P(y)=y^5+43y^4+666y^3+4\,502y^2+12\,293y+8\,415$$

si se sabe que $P_1(y)=y^2+16y+55$ es un factor cuadrático del polinomio.

www.FreeLibros.me

3.8.11 Por división sintética calcule el cociente del polinomio

$$P(y)=y^8+52y^7+1\,079y^6+11\,638y^5+71\,159y^4+252\,088y^3+503\,401y^2+513\,822y+201\,960$$

si se sabe que $P_1(y)=y^2+26y+153$ es un factor cuadrático del polinomio.

3.8.12 Utilizando el procedimiento de división sintética y sabiendo que $z_1=-2$, calcule la derivada del siguiente polinomio,

$$P(z)=z^5+26z^4+254z^3+1\,156z^2+2\,433z+1\,890$$

3.8.13 Utilizando el procedimiento de división sintética y sabiendo que las raíces de un polinomio son $z_{1-5}=[-2 \quad 4 \quad 7 \quad 11 \quad -24]$; compruebe que la derivada del polinomio original evaluado en cada raíz es la misma que la evaluación del cociente que queda al utilizar cada raíz respectivamente. El polinomio original es el siguiente,

$$P(z)=z^5+4z^4-375z^3+2\,510z^2-856z-14\,784$$

3.8.14 Por el método de deflación y división sintética (por inspección), calcule las raíces de polinomio original y del inverso. El polinomio original es el siguiente:

$$P(x)=x^5+3x^4-63x^3+13x^2+726x-1\,080$$

3.8.15 Utilizando el método de Bairstow determine el factor cuadrático del siguiente polinomio $x^5+27x^4+269x^3+1\,197x^2+2\,250x+1\,296=0$, con una aproximación inicial de $p_0=1$ y $q_0=1$.

3.8.16 Utilizando el método de Bairstow determine los factores cuadráticos del siguiente polinomio $x^6+34x^5+440x^4+2\,690x^3+7\,755x^2+9\,036x+3\,564=0$. En cada caso aplique una aproximación inicial de $p_0=1$ y $q_0=1$.

3.8.17 Utilizando el método de Bairstow determine los factores cuadráticos del siguiente polinomio $x^4+23x^3+182x^2+568x+576=0$. En cada caso aplique una aproximación inicial de $p_0=100$ y $q_0=-300$.

3.8.18 Utilizando el método de Bairstow determine los factores cuadráticos del siguiente polinomio $x^7+4x^6+6x^5+8x^4+145x^3+6\,768x^2+2\,425x+6\,454=0$. En cada caso aplique una aproximación inicial de $p_0=1$ y $q_0=1$.

3.8.19 Utilizando el método de Bairstow determine los factores cuadráticos aplicando en cada caso aplique una aproximación inicial de $p_0=2$ y $q_0=3$; el polinomio es

$$x^{10}+9x^9+54x^8+76x^7+34x^6+23x^5+567x^4+89x^3+23x^2+34x+23=0.$$

3.8.20 Utilizando el método de Laguerre determine la raíz más cercana a la condición inicial $r_0=4.5$ del siguiente polinomio:

$$x^7-6x^6-12x^5+90x^4+39x^3-324x^2-28x+240=0$$

3.8.21 Utilizando el método de Laguerre determine la raíz más cercana a la condición inicial $r_0=-10$ del siguiente polinomio:

$$x^5-x^4-27x^3+13x^2+134x-120=0$$

3.8.22 Utilizando el método de Laguerre determine la raíz más cercana a la condición inicial $r_0=200$ del siguiente polinomio:

$$x^4+184x^3+11\,836x^2+316\,704x+2\,995\,200=0$$

www.FreeLibros.me

3.8.23 Utilizando el método de Laguerre determine la raíz más cercana a la condición inicial $r_0 = 200$ del siguiente polinomio:

$$x^6 + 49x^5 + 945x^4 + 9\,035x^3 + 43\,874x^2 + 96\,216x + 60\,480 = 0$$

3.8.24 Utilizando el método de Laguerre determine las raíces del siguiente polinomio:

$$x^{10} + 103x^9 + 4\,722x^8 + 126\,798x^7 + 2\,206\,785x^6 + 25\,985\,127x^5 + 209\,407\,148x^4$$
$$+ 1\,138\,752\,932x^3 + 3\,991\,620\,384x^2 + 8\,123\,774\,400x + 7\,264\,857\,600 = 0$$

Utilice división sintética después de encontrar cada raíz para hacer *deflexión* del polinomio.

3.8.25 Utilizando el método de Bernoulli calcule la raíz de mayor módulo del polinomio $x^4 + 30x^3 + 305x^2 + 1\,200x + 1\,404 = 0$.

3.8.26 Utilizando el método de Bernoulli calcule las condiciones iniciales para hacer la primera aproximación a la raíz de mayor módulo del polinomio

$$x^4 + 27x^3 + 226x^2 + 648x + 448 = 0$$

3.8.27 Utilizando el método de Bernoulli calcule todas las raíces del siguiente polinomio propuesto:

$$x^6 + 45x^5 + 802x^4 + 7\,236x^3 + 34\,792x^2 + 84\,384x + 80\,640 = 0$$

3.8.28 Utilizando el método de Bernoulli calcule todas las raíces del siguiente polinomio propuesto:

$$x^8 + 64x^7 + 1\,708x^6 + 24\,640x^5 + 208\,054x^4 + 1\,038\,016x^3 + 2\,924\,172x^2 + 4\,098\,240x + 2\,027\,025 = 0$$

3.8.29 Utilizando el método de Bernoulli calcule las raíces del siguiente polinomio propuesto:

$$x^6 + 42x^5 + 700x^4 + 5\,880x^3 + 25\,984x^2 + 56\,448x + 46\,080 = 0$$

3.8.30 Utilizando el método de Newton calcule la raíz de menor módulo del siguiente polinomio:

$$x^6 + 11.25x^5 + 47.875x^4 + 97.5x^3 + 99.625x^2 + 48.75x + 9 = 0$$

3.8.31 Utilizando el método de Newton seguido del procedimiento de deflación, calcule todas las raíces con un máximo error de $1e^{-3}$ del siguiente polinomio:

$$x^7 + 16.3x^6 + 91.31x^5 + 233.329x^4 + 294.662x^3 + 184.72276x^2 + 53.15896x + 5.25504 = 0$$

3.8.32 Utilizando el método de Newton seguido del procedimiento de deflación, calcule las raíces con un máximo error de $1e^{-3}$ del siguiente polinomio:

$$x^5 + 2x^4 + 22x^3 + 160x^2 + 69x + 132 = 0$$

3.8.33 Utilizando el método de Newton seguido del procedimiento de deflación, calcule todas las raíces con un máximo error de $1e^{-6}$ del siguiente polinomio:

$$x^7 + 28x^6 + 322x^5 + 1\,960x^4 + 6\,769x^3 + 13\,132x^2 + 13\,068x + 504 = 0$$

3.8.34 Utilizando el método de Newton seguido del procedimiento de deflación, calcule las raíces con un máximo error de $1e^{-6}$ del siguiente polinomio:

$$x^7 + 9.21x^6 + 32.892x^5 + 57.528x^4 + 50.772x^3 + 20.502x^2 + 2.6x + 0.024 = 0$$

www.FreeLibros.me

3.8.35 Utilizando el método de diferencia de cocientes forme la tabla para aproximar las raíces del siguiente polinomio:

$$x^4+29x^3+288x^2+1116x+1\,296=0$$

3.8.36 Utilizando el método de diferencia de cocientes forme la tabla para aproximar las raíces del siguiente polinomio:

$$x^4+17x^3+101x^2+247x+210=0$$

3.8.37 Utilizando el método de diferencia de cocientes forme la tabla para aproximar las raíces del siguiente polinomio:

$$x^5-17x^4+107x^3-307x^2+396x-180=0$$

3.8.38 Utilizando el método de diferencia de cocientes forme la tabla para aproximar las raíces del siguiente polinomio:

$$x^6+59x^5+1\,330x^4+14\,290x^3+74\,129x^2+164\,851x+103\,740=0$$

3.8.39 Utilizando el método de diferencia de cocientes forme la tabla para aproximar las raíces del siguiente polinomio:

$$x^8+43x^7+784x^6+7\,882x^5+47\,509x^4+174\,307x^3+375\,066x^2+422\,568x+181\,440=0$$

3.8.40 Con el método de raíz cuadrada de Graeffe aproxime el valor absoluto de las raíces del polinomio $x^3+27x^2+234x+648=0$.

3.8.41 Con el método de raíz cuadrada de Graeffe aproxime el valor absoluto de las raíces del polinomio $x^3-5x^2-17x+21=0$.

3.8.42 Con el método de raíz cuadrada de Graeffe aproxime el valor absoluto de las raíces del siguiente polinomio:

$$x^4+34x^3+421x^2+2\,244x+4\,320=0$$

3.8.43 Con el método de raíz cuadrada de Graeffe aproxime el valor absoluto de las raíces del siguiente polinomio:

$$x^5-33x^4+406x^3-2\,262x^2+5\,353x-3\,465=0$$

3.8.44 Con el método de raíz cuadrada de Graeffe aproxime el valor absoluto de las raíces del siguiente polinomio:

$$x^5+37x^4+511x^3+3\,247x^2+9\,280x+9\,100=0$$

3.8.45 Con el método de raíz cuadrada de Graeffe aproxime el valor absoluto de las raíces del siguiente polinomio:

$$x^6+53x^5+1\,082x^4+10\,670x^3+51\,809x^2+110\,237x+68\,068=0$$

3.8.46 Con el método de Jenkins-Traub calcule las raíces del siguiente polinomio con coeficientes complejos de la forma:

$$(1+4i)x^5+(-12+56i)x^4+(35-12i)x^3+(80+34i)x^2+(3+9i)x+(77-14i)=0$$

www.FreeLibros.me

3.8.47 Con el método de Jenkins-Traub calcule las raíces del siguiente polinomio con coeficientes complejos de la forma:

$$(1+0i)x^5+(-13.999-5i)x^4+(74.99+55.998i)x^3+(-159.959-260.982i)x^2+$$
$$(1.95+463.934i)x+(150-199.95i)=0$$

3.8.48 Con el método de Jenkins-Traub calcule las raíces del siguiente polinomio con coeficientes complejos de la forma:

$$(23-4i)x^6+(-4+7i)x^5+(23+1i)x^4+(21-4i)x^3+(9+8i)x^2+(5+12i)x+(-8-5i)=0$$

3.8.49 Con el método de Jenkins-Traub calcule las raíces del siguiente polinomio con coeficientes complejos de la forma:

$$(-2.453-1.234i)x^6+(-7.434+0.277i)x^5+(-1.23-5.235i)x^4+(21.98-4.21i)x^3+$$
$$(2.679+5.348i)x^2+(0.53-5.12i)x+(-2.456-3.4565i)=0$$

www.FreeLibros.me

Capítulo 4

Solución de ecuaciones lineales simultáneas

4.1 Introducción

Existen muchos problemas físicos y numéricos cuya solución se obtiene resolviendo un conjunto de ecuaciones lineales simultáneas. El problema tiene una solución única cuando hay n ecuaciones linealmente independientes y n incógnitas. Cuando hay menos que n ecuaciones entonces no siempre se puede obtener una solución única.

Si hay más de n ecuaciones entonces existen dos posibilidades:

1. Las ecuaciones extra se pueden aislar, porque algunas son linealmente dependientes de otras. En este caso el problema se puede resolver al seleccionar las n ecuaciones que son independientes.
2. En un intento por caracterizar el error experimental, se toma un número más grande de observaciones que el necesario para determinar en forma única los parámetros de la ecuación. La técnica que resuelve este problema se considera en la sección 4.4.2.

Cuando se tienen n ecuaciones lineales simultáneas con n incógnitas, éstas se escriben en la forma siguiente:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned} \tag{4.1}$$

La notación de doble subíndice es bastante sencilla si se recuerda que el primero especifica el renglón y el segundo el número de la columna. Así, en la notación de matrices, un conjunto de coeficientes $a_{i,j}$ $(i,\ j=1,\ 2,\ldots,\ n)$ se representan con **A** y los elementos x_j y b_j $(j=1,\ 2,\ldots,\ n)$ se representan con los vectores **X** y **B**, respectivamente. Usando las reglas del álgebra matricial, el conjunto de ecuaciones se pueden representar en la forma:

$$\mathbf{AX} = \mathbf{B} \tag{4.2}$$

www.FreeLibros.me

Matemáticamente, estas ecuaciones tienen una solución única, si y sólo si el determinante de **A** es diferente de cero $(\det(\mathbf{A}) \neq 0)$. Numéricamente hay una dificultad, ya que el concepto de cero es más que impreciso en términos de computación. El resultado de un cálculo para el cual la respuesta es cero puede, por ejemplo, ser 10^{-50}, como el resultado de los datos o un error de redondeo. Es claro que un valor muy pequeño del determinante ocasionaría un problema tan parecido como si éste fuera cero, ya que ambos casos pueden ser indistinguibles. Un conjunto de ecuaciones con esta característica se llama *mal condicionado*; se caracteriza por el hecho de que un pequeño cambio en las condiciones puede causar un cambio grande en la respuesta.

Hay dos tipos de métodos que se analizarán para resolver el sistema de ecuaciones (4.1).

1. Los *métodos directos.* Aquí destacan dos metodologías. La primera se basa en la eliminación de elementos, y entre sus métodos sobresalen el método de Gauss, el método de Gauss-Jordan y el cálculo de la matriz inversa; la segunda se basa en la factorización, y entre sus métodos destacan la descomposición triangular inferior y superior, la factorización de Doolittle-Crout, el método de Cholesky y la factorización QR.
2. Los *métodos indirectos,* o *métodos iterativos,* se basan en el cálculo de una sucesión de aproximaciones que se espera converjan a un valor suficientemente cercano a la solución verdadera. Los dos más comunes son el método de Jacobi y el método de Gauss-Seidel.

Es importante que se aprecien las propiedades de estos dos tipos de métodos, ya que un método es mejor que otro dependiendo de la situación en particular. La ventaja de los métodos directos es que la cantidad de cálculos es fija y se puede determinar de antemano; por el contrario, para el caso de los métodos iterativos, los cálculos deben continuar indefinidamente hasta que las soluciones tengan una convergencia apropiada. En efecto, es posible que los métodos iterativos puedan no converger del todo.

Un factor adicional que se debe considerar es el número de coeficientes que tienen valor cero. Por ejemplo, en muchos problemas que parten de ecuaciones diferenciales ordinarias o parciales, se produce un conjunto numeroso de ecuaciones simultáneas que tienen pocos coeficientes diferentes de cero. En forma compacta, tales sistemas generan matrices de coeficientes que se llaman *matrices dispersas*. Esto incrementa el atractivo de los métodos iterativos considerablemente, pues el trabajo necesario es directamente proporcional al número de elementos diferentes de cero.

Se debe tener la precaución de que, en el caso de algunas ecuaciones dispersas con ciertas estructuras muy simples, se puede emplear un método directo en donde la cantidad de cálculos necesarios disminuye en forma significativa, pues es directamente proporcional al número de elementos no nulos. La ventaja de los métodos iterativos se pierde en este caso y la naturaleza finita de los métodos directos los hace más apropiados.

Cuando una matriz está mal condicionada numéricamente, algunos de los métodos pueden tener o no soluciones erróneas. Para evitar o minimizar este tipo de situaciones se utilizan técnicas de pivoteo, sea parcial (por columna) o total (toda la matriz). *Pivotear* no es otra cosa que escoger los valores más grandes y ponerlos como pivotes o puntos de giro de la diagonal.

4.2 Métodos directos

La base de los métodos directos es la eliminación de incógnitas. En un cálculo manual típico, esto se hace de modo aleatorio; sin embargo, se trata de encontrar el orden más fácil de eliminación de las incógnitas. Eliminar en forma aleatoria tiene la desventaja de que el orden aleatorio de las operaciones no es apropiado para el cálculo en una computadora, ya que es necesario un orden sistemático en los cálculos para que, si ocurre una falla en los cálculos, sea posible identificar la causa del problema. Una situación común, en la que considerables cálculos manuales nos llevan a la respuesta $0 \equiv 0$, se debe a que el problema se formuló incorrectamente, o a que se realizaron los cálculos en forma incorrecta.

4.2.1 Eliminación gaussiana

Inicialmente se describe un esquema simple conocido como *eliminación gaussiana* [Maron, 1995], [Burden *et al.*, 2002], [Nieves *et al.*, 2002], [Rodríguez, 2003]. Sin embargo, para que un esquema sea compu-

www.FreeLibros.me

tacionalmente correcto, debe incluir rutinas para *escalar* la matriz y para realizar un pivoteo parcial durante la eliminación. Estos refinamientos se presentan posteriormente en esta sección. En este tratamiento simple se supone que todos los divisores son no nulos. Las ecuaciones por resolver son

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & \\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n
\end{aligned} \tag{4.3}$$

La primera ecuación se almacena para usarla después, y la variable x_1 se elimina de las restantes $n-1$ ecuaciones al restar un múltiplo apropiado de la primera ecuación de cada una de las otras ecuaciones. Se usa la siguiente notación para los coeficientes originales,

$$a_{ij}^{(1)} = a_{ij}, \quad (i, j = 1,\ 2, \ldots, n) \tag{4.4}$$

y

$$b_i^{(1)} = b_i, \quad (i = 1,\ 2, \ldots, n) \tag{4.5}$$

Los nuevos coeficientes se encuentran al usar los multiplicadores

$$m_{i1} = \frac{a_{i1}^{(1)}}{a_{11}^{(1)}}, \quad (i,\ j = 2,\ 3, \ldots, n) \tag{4.6}$$

con ellos se forman los nuevos elementos, para la matriz **A**

$$a_{ij}^{(2)} = a_{ij}^{(1)} - m_{i1}a_{1j}^{(1)}, \quad (i = 2,\ 3, \ldots, n), \quad (j = 1,\ 2, \ldots, n) \tag{4.7}$$

y para el vector **B**:

$$b_i^{(2)} = b_i^{(1)} - m_{i1}b_i^{(1)}, \quad (i = 2,\ 3, \ldots, n) \tag{4.8}$$

Se puede observar que los elementos de la primera columna $j = 1$, tienen los siguientes valores:

$$a_{i1}^{(2)} = a_{i1}^{(1)} - \frac{a_{i1}^{(1)}}{a_{11}^{(1)}} a_{11}^{(1)} = 0 \tag{4.9}$$

por lo que la primera variable x_1 se ha eliminado de las últimas $n-1$ ecuaciones. Si se ignoran el primer renglón y la primera columna, las ecuaciones restantes tienen la misma forma que (4.3) pero con un renglón y una columna menos. Si se repite el procedimiento previo $n-1$ veces, la ecuación restante tendrá una sola incógnita y se puede resolver en forma directa.

En cada etapa del proceso, cuando la variable x_k se va a eliminar, los multiplicadores son

$$m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, \quad (i = k+1,\ k+2, \ldots, n) \tag{4.10}$$

y los nuevos elementos formados son, para la matriz **A**

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik}a_{kj}^{(k)}, \quad (i = k+1,\ k+2, \ldots, n), \quad (j = k,\ k+1, \ldots, n) \tag{4.11}$$

y para el vector **B**

$$b_i^{(k+1)} = b_i^{(k)} - m_{ik}b_k^{(k)}, \quad (i = k+1,\ k+2, \ldots, n) \tag{4.12}$$

www.FreeLibros.me

El resultado de este proceso de eliminación es un conjunto triangular de ecuaciones dado por

$$
\begin{aligned}
a_{11}^{(1)}x_1 + a_{12}^{(1)}x_2 + \cdots + a_{1n}^{(1)}x_n &= b_1^{(1)} \\
a_{22}^{(2)}x_2 + \cdots + a_{2n}^{(2)}x_n &= b_2^{(2)} \\
\cdots\cdots\cdots\cdots\cdots\cdots & \\
a_{nn}^{(n)}x_n &= b_n^{(n)}
\end{aligned}
\tag{4.13}
$$

Es fácil resolver estas ecuaciones por un proceso de sustitución hacia atrás o regresiva. La última ecuación tiene la solución

$$x_n = \frac{b_n^{(n)}}{a_{nn}^{(n)}} \tag{4.14}$$

Este valor se puede sustituir en la ecuación anterior para obtener x_{n-1}, etc. Al resolver las ecuaciones hacia atrás, se pueden calcular los valores de todas las variables.

Frecuentemente ocurre que varias ecuaciones se deben resolver con el mismo conjunto de coeficientes en la matriz **A**, pero con diferentes valores de **B**. En este caso, la organización correcta del método puede disminuir considerablemente el tiempo de cálculo necesario. Si las ecuaciones (4.6), (4.7), (4.10) y (4.11) proporcionan los pasos de computación para la reducción a una forma triangular, se puede verificar que estos cálculos son independientes de los valores de **B**. Después de que se ha realizado este proceso de reducción, éste no se necesita repetir, puesto que los multiplicadores m_{ij} ya están calculados. Si todos los valores **B** están inicialmente disponibles, es posible procesarlos de manera simultánea en forma similar a la reducción triangular. Por otro lado, el almacenamiento de los multiplicadores permite que posteriormente las incógnitas se obtengan con un esfuerzo mínimo. Las ecuaciones (4.8) y (4.12), sólo dependen de los términos b_i y de los multiplicadores m_{ij} que se almacenan cuando se realiza la reducción triangular. Por esto, sólo los cálculos en las ecuaciones (4.8) y (4.12) se llevan a cabo para subsecuentes valores de **B**. Es conveniente que el número de elementos cero introducidos sea exactamente igual al número de multiplicadores, por lo que en lugar de almacenar el valor cero, el espacio de almacenamiento en la computadora se usa para almacenar el valor de m_{ij}. Por ejemplo, en la primera columna, el multiplicador almacenado en la fila (i) es m_{i1} y, en general, la posición (i, j) contiene a m_{ij} para $(j < i)$. Estas ideas se ilustran con un ejemplo; el programa 4.6.1 proporciona el código Matlab de esta técnica.

EJEMPLO 4.1

Aplicar el método de eliminación gaussiana al sistema de ecuaciones,

$$
\begin{aligned}
1x_1 + 3x_2 + 4x_3 + 6x_4 &= 22 \\
2x_1 - 1x_2 + 3x_3 + 0x_4 &= 12 \\
5x_1 + 1x_2 + 0x_3 + 2x_4 &= 8 \\
0x_1 + 4x_2 - 1x_3 + 8x_4 &= 9
\end{aligned}
$$

La forma más sencilla de aplicar el método de eliminación gaussiana es escribiendo el sistema en forma matricial, pero con la matriz **A** aumentada con el vector **B**. Esto es, escribiendo el sistema como

$$
\left[\begin{array}{cccc|c}
1 & 3 & 4 & 6 & 22 \\
2 & -1 & 3 & 0 & 12 \\
5 & 1 & 0 & 2 & 8 \\
0 & 4 & -1 & 8 & 9
\end{array}\right]
$$

www.FreeLibros.me

Usando las fórmulas (4.6) a (4.8) para eliminar la primera fila, se obtienen los multiplicadores de la siguiente forma

$$m_{21} = 2$$

$$m_{31} = 5$$

$$m_{41} = 0$$

Por lo que la matriz se reduce a

$$\left[\begin{array}{cccc|c} 1 & 3 & 4 & 6 & 22 \\ 0 & -7 & -5 & -12 & -32 \\ 0 & -14 & -20 & -28 & -102 \\ 0 & 4 & -1 & 8 & 9 \end{array}\right]$$

Aplicando el mismo procedimiento, pero ahora considerando sólo la submatriz formada por las tres últimas filas y las cuatro últimas columnas se tienen los multiplicadores

$$m_{32} = 2$$

$$m_{42} = -\frac{4}{7}$$

así se obtiene

$$\left[\begin{array}{cccc|c} 1 & 3 & 4 & 6 & 22 \\ 0 & -7 & -5 & -12 & -32 \\ 0 & 0 & -10 & -4 & -38 \\ 0 & 0 & -\frac{27}{7} & \frac{8}{7} & \frac{65}{7} \end{array}\right]$$

Aplicando el mismo procedimiento a la submatriz formada por las dos últimas filas y las tres últimas columnas, se obtiene el multiplicador $m_{43} = -\frac{27}{70}$, y la matriz se reduce a

$$\left[\begin{array}{cccc|c} 1 & 3 & 4 & 6 & 22 \\ 0 & -7 & -5 & -12 & -32 \\ 0 & 0 & -10 & -4 & -38 \\ 0 & 0 & 0 & \frac{94}{35} & \frac{188}{35} \end{array}\right]$$

Esto es equivalente al sistema

$$\begin{aligned} x_1 + 3x_2 + 4x_3 + 6x_4 &= 22 \\ -7x_2 - 5x_3 - 12x_4 &= -32 \\ -10x_3 - 4x_4 &= -38 \\ \frac{94}{35}x_4 &= \frac{188}{35} \end{aligned}$$

Despejando x_4 de la cuarta ecuación, se tiene que $x_4 = 2$. Sustituyendo x_4 en la tercera ecuación y así sucesivamente, se tiene que la solución al sistema de ecuaciones es

$$x_4 = 2,\ x_3 = 3,\ x_2 = -1 \text{ y } x_1 = 1.$$

www.FreeLibros.me

Cuando se realiza la eliminación gaussiana, es conveniente guardar los multiplicadores en la misma matriz, ya que sabemos que la parte inferior de la matriz contendrá sólo ceros, por lo que la matriz quedaría transformada en

$$\begin{bmatrix} 1 & 3 & 4 & 6 \\ 2 & -7 & -5 & -12 \\ 5 & 2 & -10 & -4 \\ 0 & -\frac{4}{7} & -\frac{27}{70} & \frac{94}{35} \end{bmatrix}$$

Otra forma de guardar la información es escribiendo los multiplicadores en una matriz triangular inferior **L** con los unos en la diagonal y otra matriz triangular superior **U** con la matriz reducida. Esto es,

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 5 & 2 & 1 & 0 \\ 0 & -\frac{4}{7} & -\frac{27}{70} & 1 \end{bmatrix} \text{ y } \mathbf{U} = \begin{bmatrix} 1 & 3 & 4 & 6 \\ 0 & -7 & -5 & -12 \\ 0 & 0 & -10 & -4 \\ 0 & 0 & 0 & \frac{94}{35} \end{bmatrix}$$

Se puede notar que $\mathbf{A} = \mathbf{LU}$. Este procedimiento se conoce como la *descomposición* **LU**. Si se necesita resolver el sistema $\mathbf{AX} = \mathbf{B}$ se reescribe el sistema como $\mathbf{LUX} = \mathbf{B}$. Introduciendo una variable auxiliar **Y** se tienen las ecuaciones

$$\mathbf{UX} = \mathbf{Y} \text{ y } \mathbf{LY} = \mathbf{B}.$$

Dado un vector **B**, se despeja **Y** del sistema $\mathbf{LY} = \mathbf{B}$ y luego se despeja **X** del sistema $\mathbf{UX} = \mathbf{Y}$. Este tema se aborda en la sección 4.2.4. Un resultado importante que se deduce de esto es considerar el determinante de la matriz $\mathbf{A} = \mathbf{LU}$. Si se tiene que

$$\det(\mathbf{A}) = \det(\mathbf{LU}) = \det(\mathbf{L})\det(\mathbf{U})$$

Dado que $\det(\mathbf{L}) = 1$ se infiere que

$$\det(\mathbf{A}) = \det(\mathbf{U})$$

por lo que $\det(\mathbf{A}) = u_{11}u_{22}\cdots u_{nn}$; es decir, $\det(\mathbf{A})$ es igual al producto de los elementos de la diagonal de **U**. Esta forma de calcular el determinante de una matriz es la más económica en cuanto a costo de trabajo de computación. Para el ejemplo 4.1 se tiene que

$$\det(\mathbf{A}) = 1(-7)(-10)\left(\frac{94}{35}\right) = 188$$

EJEMPLO 4.2

Aplicar el método de eliminación gaussiana al siguiente sistema de ecuaciones

$$\begin{aligned} 4x_1 + x_2 + x_3 + x_4 &= 2 \\ x_1 + 3x_2 - x_3 + x_4 &= 4 \\ x_1 - x_2 + 2x_3 \quad\quad &= 6 \\ x_1 + x_2 \quad\quad + 2x_4 &= 8 \end{aligned}$$

www.FreeLibros.me

En forma matricial, el sistema queda de la siguiente manera

$$\begin{bmatrix} 4 & 1 & 1 & 1 \\ 1 & 3 & -1 & 1 \\ 1 & -1 & 2 & 0 \\ 1 & 1 & 0 & 2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 6 \\ 8 \end{bmatrix}$$

Aplicando el método de Gauss, se tienen las operaciones siguientes para la matriz aumentada,

$$\begin{bmatrix} 4 & 1 & 1 & 1 \\ 1 & 3 & -1 & 1 \\ 1 & -1 & 2 & 0 \\ 1 & 1 & 0 & 2 \end{bmatrix}\begin{bmatrix} 2 \\ 4 \\ 6 \\ 8 \end{bmatrix}, \begin{array}{l} \mathbf{R}_2 = \mathbf{R}_2 + \left(-\tfrac{1}{4}\right)\mathbf{R}_1 \text{, tal que } A_{21} = 0 \\ \mathbf{R}_3 = \mathbf{R}_3 + \left(-\tfrac{1}{4}\right)\mathbf{R}_1 \text{, tal que } A_{31} = 0 \\ \mathbf{R}_4 = \mathbf{R}_4 + \left(-\tfrac{1}{4}\right)\mathbf{R}_1 \text{, tal que } A_{41} = 0 \end{array}$$

Después de hacer estas operaciones se sigue con el siguiente pivote para hacer ceros en la siguiente columna, esto es

$$\begin{bmatrix} 4 & 1 & 1 & 1 \\ 0 & \frac{11}{4} & -\frac{5}{4} & \frac{3}{4} \\ 0 & -\frac{5}{4} & \frac{7}{4} & -\frac{1}{4} \\ 0 & \frac{3}{4} & -\frac{1}{4} & \frac{7}{4} \end{bmatrix}\begin{bmatrix} 2 \\ \frac{14}{4} \\ \frac{22}{4} \\ \frac{30}{4} \end{bmatrix}, \begin{array}{l} \mathbf{R}_3 = \mathbf{R}_3 + \left(-\tfrac{5}{11}\right)\mathbf{R}_2 \text{, tal que } A_{32} = 0 \\ \mathbf{R}_4 = \mathbf{R}_4 + \left(-\tfrac{3}{11}\right)\mathbf{R}_2 \text{, tal que } A_{42} = 0 \end{array}$$

De igual manera se hace para la tercera columna,

$$\begin{bmatrix} 4 & 1 & 1 & 1 \\ 0 & \frac{11}{4} & -\frac{5}{4} & \frac{3}{4} \\ 0 & 0 & \frac{13}{11} & \frac{1}{11} \\ 0 & 0 & \frac{1}{11} & \frac{17}{11} \end{bmatrix}\begin{bmatrix} 2 \\ \frac{14}{4} \\ \frac{78}{11} \\ \frac{72}{11} \end{bmatrix}, \quad \mathbf{R}_4 = \mathbf{R}_4 + \left(-\tfrac{1}{13}\right)\mathbf{R}_3 \text{, tal que } A_{43} = 0$$

Finalmente se obtiene la siguiente matriz;

$$\begin{bmatrix} 4 & 1 & 1 & 1 \\ 0 & \frac{11}{4} & -\frac{5}{4} & \frac{3}{4} \\ 0 & 0 & \frac{13}{11} & \frac{1}{11} \\ 0 & 0 & 0 & \frac{20}{13} \end{bmatrix}\begin{bmatrix} 2 \\ \frac{14}{4} \\ \frac{78}{11} \\ \frac{78}{13} \end{bmatrix}$$

La solución final del sistema se hace de la siguiente manera,

1. De la ecuación 4

$$\frac{20}{13}x_4 = \frac{78}{13},$$

Por tanto, $x_4 = \dfrac{39}{10}$

2. De la ecuación 3

$$\left(\frac{13}{11}\right)x_3 + \left(\frac{1}{11}\right)x_4 = \frac{78}{11}$$

www.FreeLibros.me

Como se conoce una de las variables, sustituyéndola en la ecuación se tiene,

$$\left(\frac{13}{11}\right)x_3+\left(\frac{1}{11}\right)\left(\frac{39}{10}\right)=\frac{78}{11},$$

En consecuencia, $x_3=\frac{57}{10}$

3. De la ecuación 2,

$$\left(\frac{11}{4}\right)x_2-\left(\frac{5}{4}\right)x_3+\left(\frac{3}{4}\right)x_4=\frac{14}{4}$$

Como se conocen dos de las variables, sustituyéndolas en la ecuación se tiene,

$$\left(\frac{11}{4}\right)x_2-\left(\frac{5}{4}\right)\left(\frac{57}{10}\right)+\left(\frac{3}{4}\right)\left(\frac{39}{10}\right)=\frac{14}{4},$$

de modo que, $x_2=\frac{14}{5}$

4. De la ecuación 1

$$4x_1+x_2+x_3+x_4=2$$

Como se conocen tres de las variables, sustituyéndolas en la ecuación se tiene,

$$4x_1+\left(\frac{14}{5}\right)+\left(\frac{57}{10}\right)+\left(\frac{39}{10}\right)=2,$$

y finalmente, $x_1=-\frac{13}{5}$

4.2.2 Eliminación de Gauss-Jordan

Existen diversas variantes del método de eliminación gaussiana;. por ejemplo, el esquema de eliminación de Jordan [Burden *et al.*, 2002], [Rodríguez, 2003]. En esta variante, la matriz, después de la eliminación tiene una forma final diagonal. Cada ecuación tiene sólo una variable, por lo que se evita el proceso de sustitución regresiva, y los valores de las variables se pueden calcular directamente.

Para aplicar el método, se procede de una manera muy similar a la eliminación gaussiana, pero en cada etapa se elimina la variable x_k, no sólo de las ecuaciones sucesivas, sino también de todas las precedentes. Las ecuaciones (4.11) y (4.12) se usan para todo $i\neq k$. La eliminación de Jordan necesita, aproximadamente, $\frac{n^3}{2}$ operaciones, en comparación con las $\frac{n^3}{3}$ de la eliminación gaussiana. No es recomendable para el uso general. El programa de cómputo desarrollado en Matlab se proporciona en la sección 4.6.2.

EJEMPLO 4.3

Aplicar el método de Gauss-Jordan al sistema del ejemplo 4.2, el cual es

$$\begin{aligned}4x_1+x_2+x_3+x_4&=2\\x_1+3x_2-x_3+x_4&=4\\x_1-x_2+2x_3\qquad&=6\\x_1+x_2\qquad+2x_4&=8\end{aligned}$$

www.FreeLibros.me

Se realiza la eliminación hasta obtener sólo una diagonal de unos que, junto con la matriz aumentada, da la solución. Partiendo de lo obtenido da la eliminación de Gauss se tiene

$$\begin{bmatrix} 4 & 1 & 1 & 1 \\ 0 & \frac{11}{4} & -\frac{5}{4} & \frac{3}{4} \\ 0 & 0 & \frac{13}{11} & \frac{1}{11} \\ 0 & 0 & 0 & \frac{20}{13} \end{bmatrix} \begin{bmatrix} 2 \\ \frac{14}{4} \\ \frac{78}{11} \\ \frac{78}{13} \end{bmatrix}, \quad \begin{aligned} \mathbf{R}_1 &= \mathbf{R}_1\left(\tfrac{1}{4}\right) \\ \mathbf{R}_2 &= \mathbf{R}_2\left(\tfrac{4}{11}\right) \\ \mathbf{R}_3 &= \mathbf{R}_3\left(\tfrac{11}{13}\right) \\ \mathbf{R}_4 &= \mathbf{R}_4\left(\tfrac{13}{20}\right) \end{aligned}$$

Al efectuar las operaciones para tener unos en la diagonal, se llega a

$$\begin{bmatrix} 1 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 0 & 1 & -\frac{5}{4} & \frac{3}{11} \\ 0 & 0 & 1 & \frac{1}{13} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{2}{4} \\ \frac{14}{11} \\ \frac{78}{13} \\ \frac{78}{20} \end{bmatrix}, \quad \begin{aligned} &\mathbf{R}_1 = \mathbf{R}_1 - \left(\tfrac{1}{4}\right)\mathbf{R}_2 \text{, tal que } A_{12} = 0 \\ &\mathbf{R}_2 = \mathbf{R}_2 + \left(\tfrac{5}{4}\right)\mathbf{R}_3 \text{, tal que } A_{23} = 0 \\ &\mathbf{R}_3 = \mathbf{R}_3 - \left(\tfrac{1}{13}\right)\mathbf{R}_4 \text{, tal que } A_{34} = 0 \end{aligned}$$

Haciendo ceros justo sobre la diagonal se obtiene

$$\begin{bmatrix} 1 & 0 & \frac{4}{11} & \frac{2}{11} \\ 0 & 1 & 0 & \frac{4}{13} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{2}{11} \\ \frac{52}{13} \\ \frac{57}{10} \\ \frac{78}{20} \end{bmatrix}, \quad \begin{aligned} &\mathbf{R}_1 = \mathbf{R}_1 - \left(\tfrac{4}{11}\right)\mathbf{R}_3 \text{, tal que } A_{13} = 0 \\ &\mathbf{R}_2 = \mathbf{R}_2 - \left(\tfrac{4}{13}\right)\mathbf{R}_4 \text{, tal que } A_{24} = 0 \end{aligned}$$

Siguiendo el procedimiento de poner ceros en la matriz, se obtiene

$$\begin{bmatrix} 1 & 0 & 0 & \frac{2}{11} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -\frac{104}{55} \\ \frac{14}{5} \\ \frac{57}{10} \\ \frac{39}{10} \end{bmatrix}, \quad \mathbf{R}_1 = \mathbf{R}_1 - \left(\tfrac{2}{11}\right)\mathbf{R}_4 \text{, tal que } A_{14} = 0$$

Finalmente se obtiene la matriz

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -\frac{13}{5} \\ \frac{14}{5} \\ \frac{57}{10} \\ \frac{39}{10} \end{bmatrix}$$

Así las soluciones son

$$x_1 = -\frac{13}{5}$$

$$x_2 = \frac{14}{5}$$

$$x_3 = \frac{57}{10}$$

$$x_4 = \frac{39}{10}$$

www.FreeLibros.me

4.2.3 Inversa de una matriz

En el caso particular de un conjunto de ecuaciones con diferentes entradas de **B**, es apropiado encontrar la solución calculando la *inversa de la matriz* **A**, para lo cual se requiere que ésta sea *no singular*. Ésta se determina al resolver **X** de la ecuación **AX** = **I**, donde la matriz **A** tiene los coeficientes del lado izquierdo de la ecuación (4.3) y el lado derecho, **I**, es la *matriz identidad* [Maron, 1995], [Burden *et al.*, 2002], [Rodríguez, 2003],

$$\mathbf{I}=\begin{bmatrix}1 & 0 & \cdots & 0\\ 0 & 1 & \cdots & 0\\ \vdots & \vdots & \cdots & \vdots\\ 0 & 0 & \cdots & 1\end{bmatrix} \tag{4.15}$$

Resolviendo cada uno de estos vectores columna de **I** se obtienen las columnas de la matriz inversa $\mathbf{A}^{-1}$. Se debe señalar que el método de los determinantes de Cramer y la fórmula

$$\mathbf{A}^{-1}=\frac{\text{adjunta}\,(\mathbf{A})}{\det(\mathbf{A})} \tag{4.16}$$

que son útiles para resolver ecuaciones, son poco apropiados para su uso en computadora. El determinante de una matriz se puede encontrar de una forma bastante fácil como producto del proceso de la eliminación gaussiana, si el proceso se efectúa como se describió antes. Así, el determinante es el producto de los elementos de la diagonal de la matriz triangular que se utiliza en el proceso de sustitución regresiva. El programa de cómputo para determinar la inversa de una matriz se presenta en la sección 4.6.3.

EJEMPLO 4.4

Aplicar el método de inversa de una matriz al sistema del ejemplo 4.2,

$$\begin{aligned}4x_1+x_2+x_3+x_4&=2\\ x_1+3x_2-x_3+x_4&=4\\ x_1-x_2+2x_3\qquad&=6\\ x_1+x_2\qquad+2x_4&=8\end{aligned}$$

El proceso para obtener la inversa de una matriz se hace como la eliminación de Gauss-Jordan, poniendo una matriz diagonal en la parte derecha. Al final lo que queda en esta matriz es la inversa.

$$\left[\begin{array}{cccc|cccc}4 & 1 & 1 & 1 & 1 & 0 & 0 & 0\\ 1 & 3 & -1 & 1 & 0 & 1 & 0 & 0\\ 1 & -1 & 2 & 0 & 0 & 0 & 1 & 0\\ 1 & 1 & 0 & 2 & 0 & 0 & 0 & 1\end{array}\right],\quad \begin{array}{l}\mathbf{R}_2=\mathbf{R}_2+\left(-\tfrac{1}{4}\right)\mathbf{R}_1\text{, tal que } A_{21}=0\\ \mathbf{R}_3=\mathbf{R}_3+\left(-\tfrac{1}{4}\right)\mathbf{R}_1\text{, tal que } A_{31}=0\\ \mathbf{R}_4=\mathbf{R}_4+\left(-\tfrac{1}{4}\right)\mathbf{R}_1\text{, tal que } A_{41}=0\end{array}$$

Realizando las operaciones se obtiene,

$$\left[\begin{array}{cccc|cccc}4 & 1 & 1 & 1 & 1 & 0 & 0 & 0\\ 0 & \tfrac{11}{4} & -\tfrac{5}{4} & \tfrac{3}{4} & -\tfrac{1}{4} & 1 & 0 & 0\\ 0 & -\tfrac{5}{4} & \tfrac{7}{4} & -\tfrac{1}{4} & -\tfrac{1}{4} & 0 & 1 & 0\\ 0 & \tfrac{3}{4} & -\tfrac{1}{4} & \tfrac{7}{4} & -\tfrac{1}{4} & 0 & 0 & 1\end{array}\right],\quad \begin{array}{l}\mathbf{R}_3=\mathbf{R}_3+\left(-\tfrac{5}{11}\right)\mathbf{R}_2\text{, tal que } A_{32}=0\\ \mathbf{R}_4=\mathbf{R}_4+\left(-\tfrac{3}{11}\right)\mathbf{R}_2\text{, tal que } A_{42}=0\end{array}$$

De igual forma se pasa a,

www.FreeLibros.me

$$\left[\begin{array}{cccc|cccc} 4 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & \frac{11}{4} & -\frac{5}{4} & \frac{3}{4} & -\frac{1}{4} & 1 & 0 & 0 \\ 0 & 0 & \frac{13}{11} & \frac{1}{11} & -\frac{4}{11} & \frac{5}{11} & 1 & 0 \\ 0 & 0 & \frac{1}{11} & \frac{17}{11} & -\frac{2}{11} & -\frac{3}{11} & 0 & 1 \end{array}\right], \quad \mathbf{R}_4 = \mathbf{R}_4 + \left(-\tfrac{1}{13}\right)\mathbf{R}_3 \text{, tal que } A_{43} = 0$$

De esta forma se obtiene

$$\left[\begin{array}{cccc|cccc} 4 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & \frac{11}{4} & -\frac{5}{4} & \frac{3}{4} & -\frac{1}{4} & 1 & 0 & 0 \\ 0 & 0 & \frac{13}{11} & \frac{1}{11} & -\frac{4}{11} & \frac{5}{11} & 1 & 0 \\ 0 & 0 & 0 & \frac{20}{13} & -\frac{2}{13} & -\frac{4}{13} & -\frac{1}{13} & 1 \end{array}\right], \quad \begin{array}{l} \mathbf{R}_1 = \mathbf{R}_1\left(\tfrac{1}{4}\right) \\ \mathbf{R}_2 = \mathbf{R}_2\left(\tfrac{4}{11}\right) \\ \mathbf{R}_3 = \mathbf{R}_3\left(\tfrac{11}{13}\right) \\ \mathbf{R}_4 = \mathbf{R}_4\left(\tfrac{13}{20}\right) \end{array}$$

Se ponen unos en la diagonal para obtener

$$\left[\begin{array}{cccc|cccc} 1 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 \\ 0 & 1 & -\frac{5}{11} & \frac{3}{11} & -\frac{1}{11} & \frac{4}{11} & 0 & 0 \\ 0 & 0 & 1 & \frac{1}{13} & -\frac{4}{13} & \frac{5}{13} & \frac{11}{13} & 0 \\ 0 & 0 & 0 & 1 & -\frac{2}{20} & -\frac{4}{20} & -\frac{1}{20} & \frac{13}{20} \end{array}\right], \quad \begin{array}{l} \mathbf{R}_1 = \mathbf{R}_1 - \left(\tfrac{1}{4}\right)\mathbf{R}_2 \text{, tal que } A_{12} = 0 \\ \mathbf{R}_2 = \mathbf{R}_2 + \left(\tfrac{5}{4}\right)\mathbf{R}_3 \text{, tal que } A_{23} = 0 \\ \mathbf{R}_3 = \mathbf{R}_3 - \left(\tfrac{1}{13}\right)\mathbf{R}_4 \text{, tal que } A_{34} = 0 \end{array}$$

Se continúa con las operaciones para obtener

$$\left[\begin{array}{cccc|cccc} 1 & 0 & \frac{4}{11} & \frac{2}{11} & \frac{3}{11} & -\frac{1}{11} & 0 & 0 \\ 0 & 1 & 0 & \frac{4}{13} & -\frac{3}{13} & \frac{7}{13} & \frac{5}{13} & 0 \\ 0 & 0 & 1 & 0 & -\frac{6}{20} & \frac{8}{20} & \frac{17}{20} & -\frac{1}{20} \\ 0 & 0 & 0 & 1 & -\frac{2}{20} & -\frac{4}{20} & -\frac{1}{20} & \frac{13}{20} \end{array}\right], \quad \begin{array}{l} \mathbf{R}_1 = \mathbf{R}_1 - \left(\tfrac{4}{11}\right)\mathbf{R}_3 \text{, tal que } A_{13} = 0 \\ \mathbf{R}_2 = \mathbf{R}_2 - \left(\tfrac{4}{13}\right)\mathbf{R}_4 \text{, tal que } A_{24} = 0 \end{array}$$

Así se llega a

$$\left[\begin{array}{cccc|cccc} 1 & 0 & 0 & \frac{2}{11} & \frac{21}{55} & -\frac{13}{55} & -\frac{17}{55} & \frac{1}{55} \\ 0 & 1 & 0 & 0 & -\frac{1}{5} & \frac{3}{5} & \frac{2}{5} & -\frac{1}{5} \\ 0 & 0 & 1 & 0 & -\frac{6}{20} & \frac{8}{20} & \frac{17}{20} & -\frac{1}{20} \\ 0 & 0 & 0 & 1 & -\frac{2}{20} & -\frac{4}{20} & -\frac{1}{20} & \frac{13}{20} \end{array}\right], \quad \mathbf{R}_1 = \mathbf{R}_1 - \left(\tfrac{2}{11}\right)\mathbf{R}_4 \text{, tal que } A_{14} = 0$$

Finalmente se obtiene

$$\left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & \frac{2}{5} & -\frac{1}{5} & -\frac{3}{10} & -\frac{1}{10} \\ 0 & 1 & 0 & 0 & -\frac{1}{5} & \frac{3}{5} & \frac{2}{5} & -\frac{1}{5} \\ 0 & 0 & 1 & 0 & -\frac{3}{10} & \frac{2}{5} & \frac{17}{20} & -\frac{1}{20} \\ 0 & 0 & 0 & 1 & -\frac{1}{10} & -\frac{1}{5} & -\frac{1}{20} & \frac{13}{20} \end{array}\right]$$

Así, la matriz inversa queda del lado derecho. Se puede notar que se efectuaron las mismas operaciones que las de eliminación de Gauss-Jordan. Si se tiene el sistema en forma matricial como

$$\mathbf{AX} = \mathbf{B},$$

premultiplicando por la inversa se obtiene,

$$\mathbf{A}^{-1}\mathbf{AX} = \mathbf{A}^{-1}\mathbf{B}$$

www.FreeLibros.me

Finalmente se llega a la solución

$$\mathbf{X} = \mathbf{A}^{-1}\mathbf{B}$$

Por tanto, para el sistema específico se tiene que

$$\mathbf{X} = \begin{bmatrix} \frac{2}{5} & -\frac{1}{5} & -\frac{3}{10} & -\frac{1}{10} \\ -\frac{1}{5} & \frac{3}{5} & \frac{2}{5} & -\frac{1}{5} \\ -\frac{3}{10} & \frac{2}{5} & \frac{17}{20} & -\frac{1}{20} \\ -\frac{1}{10} & -\frac{1}{5} & -\frac{1}{20} & \frac{13}{20} \end{bmatrix} \begin{bmatrix} 2 \\ 4 \\ 6 \\ 8 \end{bmatrix}$$

Así se llega a la solución buscada,

$$x_1 = \left(\frac{2}{5}\right)2 - \left(\frac{1}{5}\right)4 - \left(\frac{3}{10}\right)6 - \left(\frac{1}{10}\right)8 = -\frac{13}{5} \Rightarrow x_1 = -\frac{13}{5}$$

$$x_2 = -\left(\frac{1}{5}\right)2 + \left(\frac{3}{5}\right)4 + \left(\frac{2}{5}\right)6 - \left(\frac{1}{5}\right)8 = \frac{14}{5} \Rightarrow x_2 = \frac{14}{5}$$

$$x_3 = -\left(\frac{3}{10}\right)2 + \left(\frac{2}{5}\right)4 + \left(\frac{17}{20}\right)6 - \left(\frac{1}{20}\right)8 = \frac{57}{10} \Rightarrow x_3 = \frac{57}{10}$$

$$x_4 = -\left(\frac{1}{10}\right)2 - \left(\frac{1}{5}\right)4 - \left(\frac{1}{20}\right)6 + \left(\frac{13}{20}\right)8 = \frac{39}{10} \Rightarrow x_4 = \frac{39}{10}$$

4.2.3.1 Pivoteo parcial y pivoteo total

Para matrices grandes se realiza un considerable número de operaciones aritméticas y, en cada paso del proceso, los cálculos usan las cantidades obtenidas en el paso anterior. Al hacerlo así se puede incrementar el error y, por tanto, es importante tomar todas las precauciones para minimizar este incremento [Burden *et al.*, 2002]. De las ecuaciones (4.11) y (4.12), se observa que una de las operaciones recurrente es la multiplicación por m_{ij}.

Si al multiplicar los números se tiene algún error acumulado, éste también se multiplica por m_{ij}; por tanto, estos multiplicadores se deben hacer tan pequeños como sea posible, y por supuesto menores que uno. De esta manera, los errores no se incrementan al efectuar las multiplicaciones. Esto se obtiene fácilmente si el elemento pivote $a_{kk}^{(k)}$ es el más grande de todos los elementos $a_{ik}^{(k)}$ en la misma columna k para $(i \geq k)$, entonces

$$|m_{ij}| \leq 1, \quad (i = 1, 2, \ldots, n), \quad (j < i) \tag{4.17}$$

Para implementar esta sugerencia se necesita un paso extra; se trata de un cálculo muy pequeño. En la etapa del proceso donde la siguiente x_k se va a eliminar, se realiza una búsqueda para encontrar el elemento de módulo más grande en la columna guía, por debajo del elemento pivote. La fila que contiene este elemento se intercambia con la fila pivote, por lo que el elemento más grande está ahora en la posición pivote. Esto nos da multiplicadores de módulo menor que uno. Esto se hace en cada etapa del proceso, intercambiando donde sea necesario. Este proceso de intercambio puede evitar el problema que podría ocurrir si un elemento pivote $a_{kk}^{(k)}$ fuera cero. A menos que la matriz sea singular, o cercanamente singular, el proceso de búsqueda encontrará al menos un elemento diferente a cero en cada columna, de tal manera que no ocurrirá el problema de la división entre cero. Esto se debe realizar, pues este proceso de pivoteo parcial es esencial para mantener la precisión si la matriz está mal condicionada. Este proceso se emplea en el programa de Matlab de la sección 4.6.4 de este capítulo.

También es posible extender esta idea y aplicar el pivoteo completo. Este refinamiento consiste en la búsqueda en las restantes submatrices para poner los elementos de módulos más grandes en la posición

www.FreeLibros.me

pivote. Esto implica no sólo alteración en el orden de las filas, sino también en el orden de las variables en la ecuación. En la sección 4.6.5 se da un ejemplo de lo anterior y el programa de cómputo desarrollado en Matlab.

4.2.3.2 Escalamiento

Con una pequeña reflexión se puede ver que la estrategia de pivoteo parcial por sí misma es inadecuada. Si se consideran las dos ecuaciones

$$4x+3y=10 \tag{4.18a}$$

$$3x-2y=12, \tag{4.18b}$$

el elemento pivote, con el módulo más grande, está en su lugar. Sin embargo, la simple operación de multiplicar la segunda ecuación por 2 nos indicaría que las dos ecuaciones se tienen que intercambiar para poner el elemento de módulo mayor en la posición pivote [Burden *et al.*, 2002]. La naturaleza arbitraria de la elección de una fila pivote es un obstáculo para el desarrollo del procedimiento de eliminación más apropiado. La solución para este problema es escalar la matriz de manera que las filas sean comparables de alguna forma definida. Esto se realiza usualmente mediante la normalización en una de dos formas. Las filas se pueden normalizar al dividir cada una de las filas entre el elemento de las columnas que tenga el módulo mayor, por lo que el elemento más grande de la nueva fila es uno; o alternativamente, cada fila se divide entre el cuadrado de su norma, es decir entre

$$d_i=\sqrt{\left(\sum_{j=1}^{n}a_{ij}^2\right)} \tag{4.19}$$

Al reflexionar se observa que el escalamiento puede representar una diferencia significativa en la precisión de las soluciones, y que no hay un método estándar de escalamiento universalmente aceptado.

4.2.3.3 Mal condicionamiento

Por supuesto es posible que, en matrices mal condicionadas, el esquema de pivoteo parcial falle; pero el esquema de eliminación gaussiana puede dar alguna indicación del tipo de mal condicionamiento que está causando el problema. Esto se puede ilustrar por las siguientes ecuaciones con dos variables. Multiplicando la primera ecuación por (-2) y sumando ambas ecuaciones se obtiene:

$$\begin{array}{ll} 2x+3y=10 \quad \times(-2) & \qquad 2x+3y=10 \quad \times(-2) \\ \underline{4x+6y=20} & \qquad \underline{4x+6y=22} \\ \quad 0=0 & \qquad \quad 0=2 \end{array} \tag{4.20}$$

En el primer ejemplo, el proceso de eliminación podría mostrar un elemento muy pequeño no sólo en la posición pivote, sino también en el lado derecho de la ecuación. Esto podría indicar que las n ecuaciones simultáneas no son linealmente independientes. En el segundo caso, el lado derecho no es cero. Esto indica que las ecuaciones son inconsistentes y que se cometió algún error en la formulación del problema. Está claro que en cada etapa se debe revisar el elemento pivote y se tiene que realizar el paso apropiado si el tamaño del elemento pivote es menor que algún valor pequeño.

4.2.3.4 Mejoramiento de la solución

Como una precaución elemental, cuando se ha encontrado la solución del conjunto de ecuaciones, los valores se deben sustituir en las ecuaciones originales para comprobar que las satisfacen. Si hay errores grandes, entonces las soluciones son no satisfactorias. Desafortunadamente, con ecuaciones mal condicionadas se pueden encontrar errores muy pequeños después de las sustituciones, aunque cada una de las soluciones calculadas difiera bastante de la solución verdadera. Sean las soluciones calculadas

www.FreeLibros.me

$$\mathbf{X}^{(0)} \equiv \left[x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)}\right]^T \tag{4.21}$$

Si se forman los residuales

$$\mathbf{R}^{(0)} \equiv \left[r_1^{(0)}, r_2^{(0)}, \ldots, r_n^{(0)}\right]^T \tag{4.22}$$

al sustituir en las ecuaciones (4.3) en forma matricial, se tiene

$$\mathbf{AX}^{(0)} - \mathbf{B} = \mathbf{R}^{(0)} \tag{4.23}$$

Los residuales $\mathbf{R}^{(0)}$ indicarían cuando las soluciones son inaceptables y conducen a un método bastante efectivo para mejorar la solución. Además, como la solución verdadera satisface

$$\mathbf{AX} - \mathbf{B} = \mathbf{0} \tag{4.24}$$

entonces, restando la ecuación (4.23) de la (4.24) queda

$$\mathbf{A}\left(\mathbf{X} - \mathbf{X}^{(0)}\right) = \mathbf{R}^{(0)} \tag{4.25}$$

La cantidad $\left(\mathbf{X} - \mathbf{X}^{(0)}\right)$ es la que se debe sumar a la primera solución calculada $\mathbf{X}^{(0)}$ para obtener la solución correcta. Si esto se puede calcular en forma exacta el problema está completamente resuelto. De hecho $\left(\mathbf{X} - \mathbf{X}^{(0)}\right)$, se encuentra mediante la resolución de ecuaciones simultáneas lineales. Por otro lado, se debe notar que el trabajo adicional es muy pequeño, porque la reducción a la forma triangular y el almacenamiento de los multiplicadores ya se han realizado.

Al resolver la ecuación (4.25) y al usar esta solución para corregir el valor $\left(\mathbf{X} - \mathbf{X}^{(0)}\right)$ se incrementa notablemente la precisión; cuando este incremento es importante, se recomienda esta solución.. Formalmente, si $\mathbf{E}^{(0)}$ es la solución de $\mathbf{AE} = -\mathbf{R}^{(0)}$ y la nueva aproximación de la solución es $\mathbf{X}^{(1)} = \mathbf{X}^{(0)} + \mathbf{E}^{(0)}$, se pueden hacer otras mejoras al formar los residuales

$$\mathbf{AX}^{(p)} - \mathbf{B} = \mathbf{R}^{(p)} \tag{4.26}$$

y entonces se resuelve una sucesión de ecuaciones de la forma

$$\mathbf{AE} = -\mathbf{R}^{(p)}, \quad (p = 1, 2, \ldots, n) \tag{4.27}$$

Las soluciones $\mathbf{E}^{(p)}$ de estas ecuaciones dan las nuevas aproximaciones a las soluciones,

$$\mathbf{X}^{(p+1)} = \mathbf{X}^{(p)} + \mathbf{E}^{(p)} \tag{4.28}$$

4.2.4 Factorización LU

Existe otro grupo de métodos directos que se pueden describir bajo el encabezado general de descomposición triangular; estos incluyen las *variantes de Crout, Doolittle* y *Choleski*. El esquema computacional es una manera diferente de la eliminación gaussiana, aunque los esquemas son muy similares. Estos métodos se basan en una serie de multiplicadores que reducen la matriz a una forma triangular, seguida por el proceso de sustitución hacia adelante o progresiva, y hacia atrás o regresiva, [Burden *et al.*, 2002], [Rodríguez, 2003]. El esquema supone que la descomposición triangular es teóricamente posible; es decir, que la matriz se puede expresar como el producto de dos matrices:

$$\mathbf{A} = \mathbf{LU} \tag{4.29}$$

donde **U** es una matriz triangular superior y **L** es una matriz triangular inferior. Una vez que estas matrices se han encontrado, el conjunto de ecuaciones se resuelve en dos etapas. Cada una de éstas implica la solución de un conjunto de ecuaciones con una matriz triangular. Esto es fácil de realizar mediante un proceso de sustitución progresiva o regresiva.

www.FreeLibros.me

Las ecuaciones serían:

$$\mathbf{LUX} = \mathbf{B} \tag{4.30}$$

si se encuentra un vector **Y** tal que **LY** = **B** y se resuelve por sustitución progresiva para encontrar el vector **Y**, después se usa el sistema de ecuaciones **UX** = **Y** y se resuelve por sustitución regresiva para encontrar la variable **X**, que es la incógnita buscada. El método se ilustra por medio de un conjunto de ecuaciones de dimensión 3. Aquí la matriz **U** es la matriz triangular superior que resulta al aplicar el proceso de Gauss, sin normalizar los pivotes. Así se deben encontrar los coeficientes de manera que

$$\begin{pmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{pmatrix}\begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \tag{4.31}$$

Los valores diagonales de **L** son iguales a 1, por tanto, el sistema de ecuaciones resultante es:

$$l_{21}u_{11} = a_{21}, \quad \therefore\ l_{21} = \frac{a_{21}}{u_{11}} \quad \text{(2o. renglón por 1a. columna)} \tag{4.32a}$$

$$l_{31}u_{11} = a_{31}, \quad \therefore\ l_{31} = \frac{a_{31}}{u_{11}} \quad \text{(3er. renglón por 1a. columna)} \tag{4.32b}$$

$$l_{31}u_{12} + l_{32}u_{22} = a_{32}, \quad \therefore\ l_{32} = \frac{(a_{32} - l_{31}u_{12})}{u_{22}} \quad \text{(3er. renglón por 2a. columna)} \tag{4.32c}$$

Este algoritmo utiliza una cantidad de operaciones igual que el método de eliminación gaussiana. Por este motivo se recomienda este método para el uso general donde sea necesario algún método directo. En la sección 4.6.6 se presenta el código Matlab de este método.

EJEMPLO 4.5

Aplicar el método de factorización **A** = **LU** para resolver el sistema del ejemplo 4.1, que tiene una representación matricial

$$\begin{bmatrix} 4 & 1 & 1 & 1 \\ 1 & 3 & -1 & 1 \\ 1 & -1 & 2 & 0 \\ 1 & 1 & 0 & 2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 6 \\ 8 \end{bmatrix}$$

La matriz **U** es la que nos queda después de hacer la eliminación de Gauss. Por tanto, el producto **LU** = **A** es

$$\begin{bmatrix} L_{11} & L_{12} & L_{13} & L_{14} \\ L_{21} & L_{22} & L_{23} & L_{24} \\ L_{31} & L_{32} & L_{33} & L_{34} \\ L_{41} & L_{42} & L_{43} & L_{44} \end{bmatrix}\begin{bmatrix} 4 & 1 & 1 & 1 \\ 0 & \frac{11}{4} & -\frac{5}{4} & \frac{3}{4} \\ 0 & 0 & \frac{13}{11} & \frac{1}{11} \\ 0 & 0 & 0 & \frac{20}{13} \end{bmatrix} = \begin{bmatrix} 4 & 1 & 1 & 1 \\ 1 & 3 & -1 & 1 \\ 1 & -1 & 2 & 0 \\ 1 & 1 & 0 & 2 \end{bmatrix}$$

En el ejemplo 4.2 se muestra cómo al guardar los operadores de la eliminación se puede formar la matriz triangular inferior. Otro proceso alterno es simplemente hacer las operaciones indicadas sabiendo de antemano que la matriz **L** es triangular inferior con unos en la diagonal. Se inicia con los renglones dos al cuatro y la primera columna para obtener

$$\mathbf{R}_2\mathbf{C}_1 = A_{21},\ L_{21}(4) = 1,\ L_{21} = \frac{1}{4}$$

www.FreeLibros.me

$$\mathbf{R}_3\mathbf{C}_1 = A_{31},\ L_{31}(4) = 1,\ L_{31} = \frac{1}{4}$$

$$\mathbf{R}_4\mathbf{C}_1 = A_{41},\ L_{41}(4) = 1,\ L_{41} = \frac{1}{4}$$

Se sigue con los renglones tres y cuatro y la segunda columna,

$$\mathbf{R}_3\mathbf{C}_2 = A_{32},\ L_{31}(1) + L_{32}\left(\frac{11}{4}\right) = -1,\ L_{32} = -\frac{5}{11}$$

$$\mathbf{R}_4\mathbf{C}_2 = A_{42},\ L_{41}(1) + L_{42}\left(\frac{11}{4}\right) = 1,\ L_{42} = \frac{3}{11}$$

Finalmente, el renglón cuatro y la columna tres dan el último valor buscado,

$$\mathbf{R}_4\mathbf{C}_3 = A_{43},\ L_{41}(1) + L_{42}\left(-\frac{5}{4}\right) + L_{43}\left(\frac{13}{11}\right) = 0,\ L_{43} = \frac{1}{13}$$

Así, la matriz triangular inferior es

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ L_{21} & 1 & 0 & 0 \\ L_{31} & L_{32} & 1 & 0 \\ L_{41} & L_{42} & L_{43} & 1 \end{bmatrix}$$

El sistema transformado a resolver $\mathbf{LUX} = \mathbf{B}$ queda de la siguiente manera:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{4} & 1 & 0 & 0 \\ \frac{1}{4} & -\frac{5}{11} & 1 & 0 \\ \frac{1}{4} & \frac{3}{11} & \frac{1}{13} & 1 \end{bmatrix} \begin{bmatrix} 4 & 1 & 1 & 1 \\ 0 & \frac{14}{4} & -\frac{5}{4} & \frac{3}{4} \\ 0 & 0 & \frac{13}{11} & \frac{1}{11} \\ 0 & 0 & 0 & \frac{20}{13} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 6 \\ 8 \end{bmatrix}$$

Utilizando en forma inicial una variable auxiliar $\mathbf{UX} = \mathbf{Y}$ como paso intermedio, se obtiene

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{4} & 1 & 0 & 0 \\ \frac{1}{4} & -\frac{5}{11} & 1 & 0 \\ \frac{1}{4} & \frac{3}{11} & \frac{1}{13} & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 6 \\ 8 \end{bmatrix}$$

La solución de este sistema se hace por sustitución progresiva, es decir, primero se resuelve la primera ecuación y después la segunda, y así sucesivamente, para obtener

$$y_1 = 2,\ y_2 = \frac{14}{4},\ y_3 = \frac{78}{11} \text{ y } y_4 = \frac{78}{13}.$$

De esta forma el sistema final por resolver queda de la siguiente manera

$$\begin{bmatrix} 4 & 1 & 1 & 1 \\ 0 & \frac{14}{4} & -\frac{5}{4} & \frac{3}{4} \\ 0 & 0 & \frac{13}{11} & \frac{1}{11} \\ 0 & 0 & 0 & \frac{20}{13} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 2 \\ \frac{14}{4} \\ \frac{78}{11} \\ \frac{78}{13} \end{bmatrix}$$

Este sistema se resuelve por sustitución regresiva. Con esto se obtiene la solución buscada como

$$x_4 = \frac{39}{10}$$

www.FreeLibros.me

$$x_3 = \frac{57}{10}$$
$$x_2 = \frac{14}{5}$$
$$x_1 = -\frac{13}{5}$$

4.2.5 Factorización Doolittle-Crout

En este método se factoriza la matriz **A** en una triangular inferior **D** y en una triangular superior **C**, con unos en la diagonal, que se obtiene haciendo una eliminación Gauss-Jordan, quedando finalmente $\mathbf{DC} = \mathbf{A}$ [Burden *et al.*, 2002], [Nieves *et al.*, 2002]. Este proceso se ilustra con un ejemplo, y en la sección 4.6.7 se proporciona el código desarrollado en Matlab.

EJEMPLO 4.6

Aplicar el método de factorización Doolittle-Crout a la matriz resultante de compactar el sistema de ecuaciones del ejemplo 4.1,

$$\mathbf{A} = \begin{bmatrix} 4 & 1 & 1 & 1 \\ 1 & 3 & -1 & 1 \\ 1 & -1 & 2 & 0 \\ 1 & 1 & 0 & 2 \end{bmatrix}$$

La eliminación de Gauss-Jordan hasta el punto donde se ponen unos en la diagonal lleva a

$$\mathbf{C} = \begin{bmatrix} 1 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 0 & 1 & -\frac{5}{11} & \frac{3}{11} \\ 0 & 0 & 1 & \frac{1}{13} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Si $\mathbf{DC} = \mathbf{A}$, por tanto, se tiene que

$$\begin{bmatrix} D_{11} & 0 & 0 & 0 \\ D_{21} & D_{22} & 0 & 0 \\ D_{31} & D_{32} & D_{33} & 0 \\ D_{41} & D_{42} & D_{43} & D_{44} \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 0 & 1 & -\frac{5}{11} & \frac{3}{11} \\ 0 & 0 & 1 & \frac{1}{13} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 1 & 1 & 1 \\ 1 & 3 & -1 & 1 \\ 1 & -1 & 2 & 0 \\ 1 & 1 & 0 & 2 \end{bmatrix}$$

Multiplicando los cuatro renglones por la primera columna se obtienen las siguientes ecuaciones:

$$\mathbf{R}_1\mathbf{C}_1 = A_{11},\ D_{11}(1) = 4,\ D_{11} = 4$$

$$\mathbf{R}_2\mathbf{C}_1 = A_{21},\ D_{21}(1) = 1,\ D_{21} = 1$$

$$\mathbf{R}_3\mathbf{C}_1 = A_{31},\ D_{13}(1) = 1,\ D_{31} = 1$$

$$\mathbf{R}_4\mathbf{C}_1 = A_{41},\ D_{41}(1) = 1,\ D_{41} = 1$$

www.FreeLibros.me

Para la columna dos sólo se necesitan calcular tres valores. Así se definen las ecuaciones que utilizan los valores ya calculados correspondientes a la primera columna,

$$\mathbf{R}_2\mathbf{C}_2 = A_{22},\ D_{21}\left(\frac{1}{4}\right)+D_{22}(1)=3,\ D_{22}=\frac{11}{4}$$

$$\mathbf{R}_3\mathbf{C}_2 = A_{32},\ D_{31}\left(\frac{1}{4}\right)+D_{32}(1)=-1,\ D_{32}=-\frac{5}{4}$$

$$\mathbf{R}_4\mathbf{C}_2 = A_{42},\ D_{41}\left(\frac{1}{4}\right)+D_{42}(1)=1,\ D_{42}=\frac{3}{4}$$

En la columna tres sólo hay dos incógnitas. Éstas se resuelven de las ecuaciones

$$\mathbf{R}_3\mathbf{C}_3 = A_{33},\ D_{31}\left(\frac{1}{4}\right)+D_{32}\left(-\frac{5}{11}\right)+D_{33}(1)=2,\ D_{33}=\frac{13}{11}$$

$$\mathbf{R}_4\mathbf{C}_3 = A_{43},\ D_{41}\left(\frac{1}{4}\right)+D_{42}\left(-\frac{5}{11}\right)+D_{43}(1)=0,\ D_{43}=\frac{1}{11}$$

Por último, para la columna cuatro se tiene una incógnita en función de valores de columnas anteriores previamente calculados:

$$\mathbf{R}_4\mathbf{C}_4 = A_{44},\ D_{41}\left(\frac{1}{4}\right)+D_{42}\left(\frac{3}{11}\right)+D_{43}\left(\frac{1}{13}\right)+D_{44}(1)=2,\ D_{44}=\frac{20}{13}$$

Así, se tiene la factorización final $\mathbf{DC}=\mathbf{A}$. En forma expandida será entonces:

$$\begin{bmatrix} 4 & 0 & 0 & 0 \\ 1 & \frac{11}{4} & 0 & 0 \\ 1 & -\frac{5}{4} & \frac{13}{11} & 0 \\ 1 & \frac{3}{4} & \frac{1}{11} & \frac{20}{13} \end{bmatrix}\begin{bmatrix} 1 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 0 & 1 & -\frac{5}{11} & \frac{3}{11} \\ 0 & 0 & 1 & \frac{1}{13} \\ 0 & 0 & 0 & 1 \end{bmatrix}=\begin{bmatrix} 4 & 1 & 1 & 1 \\ 1 & 3 & -1 & 1 \\ 1 & -1 & 2 & 0 \\ 1 & 1 & 0 & 2 \end{bmatrix}$$

El sistema por resolver es

$$\begin{bmatrix} 4 & 0 & 0 & 0 \\ 1 & \frac{11}{4} & 0 & 0 \\ 1 & -\frac{5}{4} & \frac{13}{11} & 0 \\ 1 & \frac{3}{4} & \frac{1}{11} & \frac{20}{13} \end{bmatrix}\begin{bmatrix} 1 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 0 & 1 & -\frac{5}{11} & \frac{3}{11} \\ 0 & 0 & 1 & \frac{1}{13} \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}=\begin{bmatrix} 2 \\ 4 \\ 6 \\ 8 \end{bmatrix}$$

Igual que en la factorización anterior, se utiliza un paso intermedio de la forma

$$\begin{bmatrix} 4 & 0 & 0 & 0 \\ 1 & \frac{11}{4} & 0 & 0 \\ 1 & -\frac{5}{4} & \frac{13}{11} & 0 \\ 1 & \frac{3}{4} & \frac{1}{11} & \frac{20}{13} \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}=\begin{bmatrix} 2 \\ 4 \\ 6 \\ 8 \end{bmatrix}$$

Por sustitución hacia adelante se obtiene la solución

$$y_1=\frac{2}{4}$$

$$y_2=\frac{14}{11}$$

www.FreeLibros.me

$$y_3 = \frac{78}{13}$$

$$y_4 = \frac{78}{20}$$

De esta forma se llega a

$$\begin{bmatrix} 1 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 0 & 1 & -\frac{5}{11} & \frac{3}{11} \\ 0 & 0 & 1 & \frac{1}{13} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \frac{2}{4} \\ \frac{14}{11} \\ \frac{78}{13} \\ \frac{39}{10} \end{bmatrix}$$

La solución buscada se obtiene por sustitución regresiva. Ésta es

$$x_4 = \frac{39}{10}$$

$$x_3 = \frac{57}{10}$$

$$x_2 = \frac{14}{5}$$

$$x_1 = -\frac{13}{5}$$

La utilización de un mismo ejemplo es para comparar la solución que producen los métodos directos y por factorización. Observando los resultados, se puede decir que para el caso de un sistema de pocas ecuaciones todos los métodos utilizados son igualmente exactos.

Otra forma de visualizar esta factorización es considerar la factorización **LU** de **A**. Definamos la matriz $\mathbf{D}_1$ como la matriz diagonal cuyos elementos son los de la diagonal de **U**. Dado que $\mathbf{A} = \mathbf{LU}$ entonces $\mathbf{A} = \mathbf{LD_1D_1}^{-1}\mathbf{U}$. Definiendo $\mathbf{D} = \mathbf{LD_1}$ y $\mathbf{C} = \mathbf{D_1}^{-1}\mathbf{U}$ se obtienen los mismos resultados. De este resultado se puede observar que esta factorización existe, si y sólo si los elementos de la diagonal de **U** son diferentes de cero. Esto es equivalente a decir que esta factorización existe, si y sólo si el determinante de **A** es no cero.

4.2.6 Método de Choleski

En el caso de una matriz simétrica, es posible reducir la cantidad de cálculos y de almacenamiento al tomar ventaja de la simetría. Si los elementos de **L** y **U** son iguales, entonces $\mathbf{U} = \mathbf{L}^T$ y se necesita calcular y almacenar sólo los elementos de **L**. Este método se conoce como *factorización de Choleski* [Maron, 1995], [Burden *et al.*, 2002], [Nieves *et al.*, 2002]. Para el primer pivote se tiene

$$C_h(1, 1) = \sqrt{A(1, 1)} \qquad \textbf{(4.33)}$$

y para la primera columna

$$C_h(2:n, 1) = \frac{A(2:n, 1)}{A(1, 1)} \qquad \textbf{(4.34)}$$

Los siguientes pivotes son:

$$C_h(i, i) = \left[A(i, i) - \sum_{k=1}^{i-1} \{C_h(i, k)\}^2 \right] \qquad \text{para} \qquad i = 2, 3, \ldots, n \qquad \textbf{(4.35)}$$

www.FreeLibros.me

Las siguientes columnas se calculan con la fórmula:

$$C_h(i, j) = \frac{1}{C_h(i, i)}\left[A(i, j) - \sum_{k=1}^{j-1} C_h(i, k) C_h(j, k)\right] \quad \textbf{(4.36)}$$

NOTA: El método de Cholesky da los factores de la forma $\mathbf{A} = \mathbf{CC}^T$ por lo que sólo se calcula la matriz triangular inferior. En cuanto al orden de cálculo, se calcula el pivote y después todos los elementos de su columna, del pivote hacia abajo. La razón es que del pivote hacia arriba los elementos son cero. También se debe observar que si la matriz **A** es definida positiva, entonces todos los elementos de la matriz **C** serán reales. Tales matrices surgen de problemas físicos y en la solución de problemas de mínimos cuadrados. Una matriz se llama *matriz definida positiva* si para cada vector no nulo **X** se tiene que $\mathbf{X^T AX} > \mathbf{0}$. La sección 4.6.8 proporciona el código Matlab de este método.

EJEMPLO 4.7

Utilizando el método de Cholesky, factorizar. Obtener los factores de la matriz triangular inferior de

$$\mathbf{A} = \begin{bmatrix} 29 & 5 & 8 & 9 & 1 \\ 3 & 61 & 8 & 12 & 4 \\ 5 & 7 & 93 & 6 & 32 \\ 3 & 1 & 34 & 65 & 2 \\ 1 & 5 & 6 & 3 & 24 \end{bmatrix}$$

La matriz triangular inferior dada por el método de Cholesky queda de la siguiente manera;

$$\mathbf{C}_{\text{inferior}} = \begin{bmatrix} 5.3852 & 0 & 0 & 0 & 0 \\ 0.9285 & 7.7549 & 0 & 0 & 0 \\ 1.4856 & 0.8537 & 9.4902 & 0 & 0 \\ 1.6713 & 1.3473 & 0.2494 & 7.7672 & 0 \\ 0.1857 & 0.4936 & 3.2984 & 0.0260 & 3.5835 \end{bmatrix}$$

El procedimiento sólo da la matriz triangular inferior; la superior es simplemente la transpuesta. Así se llega a

$$\mathbf{C}_{\text{superior}} = \begin{bmatrix} 5.3852 & 0.9285 & 1.4856 & 1.6713 & 0.1857 \\ 0 & 7.7549 & 0.8537 & 1.3473 & 0.4936 \\ 0 & 0 & 9.4902 & 0.2494 & 3.2984 \\ 0 & 0 & 0 & 7.7672 & 0.0260 \\ 0 & 0 & 0 & 0 & 3.5835 \end{bmatrix}$$

Se puede comprobar de manera fácil que, efectivamente, $\mathbf{C}_{\text{inferior}}\mathbf{C}_{\text{superior}} = \mathbf{A}$, por lo que sólo se necesita calcular y almacenar la matriz triangular inferior.

4.2.7 Factorización LU y QR

La factorización **LU** se obtiene del proceso computacional que emplea la factorización repetida de una secuencia de matrices de la forma triangular izquierda y derecha. Se supone para este propósito que es posible hacer la descomposición en forma triangular de la forma:

$$\mathbf{A} = \mathbf{A}_1 = \mathbf{L}_1\mathbf{U}_1 \quad \textbf{(4.37)}$$

www.FreeLibros.me

la siguiente matriz resulta de la multiplicación invertida de la factorización y tiene la forma:

$$\mathbf{A_2} = \mathbf{U_1 L_1} = \mathbf{L_2 U_2} \tag{4.38}$$

similarmente se tiene

$$\mathbf{A_r} = \mathbf{U_{r-1} L_{r-1}} = \mathbf{L_r U_r}, \qquad r = 1, 2, \ldots \tag{4.39}$$

La sustitución de las matrices obtenidas se efectúa así:

$$\mathbf{A_2} = (\mathbf{U_1})\mathbf{L_1} = \mathbf{L_1^{-1} A_1 L_1} \tag{4.40}$$

De forma similar,

$$\mathbf{A_3} = \mathbf{L_2^{-1} A_2 L_2} = \mathbf{L_2^{-1} L_1^{-1} A_1 L_1 L_2} \tag{4.41}$$

Esta secuencia de matrices converge a un bloque de forma triangular superior. El segundo método incorpora transformaciones ortogonales dentro del proceso que dan buenas condiciones de estabilidad. Las matrices se descomponen en un producto $\mathbf{Q_r R_r}$, donde $\mathbf{R_r}$ es triangular superior [Cordero, 2006]. Así,

$$\mathbf{A} = \mathbf{A_1} = \mathbf{Q_1 R_1} \tag{4.42}$$

$$\mathbf{A_r} = \mathbf{R_{r-1} Q_{r-1}} = \mathbf{Q_r R_r}, \qquad r = 2, 3, \ldots \tag{4.43}$$

Observe que las matrices son similares debido a que

$$\mathbf{A_r} = \mathbf{R_{r-1} Q_{r-1}} = \mathbf{Q_{r-1}^{-1} A_{r-1} Q_{r-1}} \tag{4.44}$$

La descomposición básica $\mathbf{A} = \mathbf{Q_r R_r}$ se puede realizar para cualquier matriz, lo que no es el caso de la descomposición **LU** La sección 4.6.9 proporciona el código Matlab de este método.

EJEMPLO 4.8

Utilizando el método de factorización $\mathbf{A} = \mathbf{QR}$, resolver el sistema de ecuaciones dado por $\mathbf{AX} = \mathbf{B}$, con las siguientes matrices;

$$\mathbf{A} = \begin{bmatrix} 9 & 5 & 8 & 9 & 1 \\ 3 & 6 & 8 & 2 & 4 \\ 5 & 7 & 3 & 6 & 2 \\ 3 & 1 & 4 & 5 & 2 \\ 1 & 5 & 6 & 3 & 4 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 2 \\ 7 \\ 3 \\ 8 \\ 1 \end{bmatrix}$$

El método **QR** es una transformación donde se llega a una matriz triangular superior. Así, utilizando la fórmula (4.44), después de la enésima iteración se obtiene una matriz triangular superior de la forma

$$\mathbf{A_S} = \mathbf{Q_n^{-1} Q_{n-1}^{-1} \cdots Q_2^{-1} Q_1^{-1} A Q_1 Q_2 \cdots Q_{n-1} Q_n}$$

Si se agrupan las matrices como $\mathbf{Q_i} = \mathbf{Q_n^{-1} Q_{n-1}^{-1} \cdots Q_2^{-1} Q_1^{-1}}$ y $\mathbf{Q_d} = \mathbf{Q_1 Q_2 \cdots Q_{n-1} Q_n}$, entonces se tiene que la matriz **A** se puede representar como

$$\mathbf{A} = \mathbf{Q_d A_S Q_i}$$

Para calcular la inversa de la matriz **A**, basta invertir la matriz triangular superior $\mathbf{A_S}$

www.FreeLibros.me

$$\mathbf{A}^{-1} = \mathbf{Q_d A_S^{-1} Q_i}$$

Por tanto, la solución del sistema $\mathbf{AX} = \mathbf{B}$, queda de la siguiente manera

$$\mathbf{X} = \mathbf{Q_d A_S^{-1} Q_i B}$$

Para el ejercicio de aplicación se llega a una matriz triangular superior $\mathbf{A_S}$ con la siguiente estructura:

$$\mathbf{A_S} = \begin{bmatrix} 22.5105 & 0.7552 & 2.3745 & 3.5919 & -6.6002 \\ 0 & 6.4587 & 1.9442 & -1.9107 & 0.9681 \\ 0 & 0 & -4.0855 & -0.4524 & -1.3183 \\ 0 & 0 & 0 & 2.7366 & -0.1278 \\ 0 & 0 & 0 & 0 & -0.6201 \end{bmatrix}$$

El producto $\mathbf{Q_d A_S^{-1} Q_i}$, lo que es igual a tener $\mathbf{A}^{-1}$, es

$$\mathbf{Q_d A_S^{-1} Q_i} = \begin{bmatrix} -0.2143 & 0.5000 & 0.1984 & 0.3929 & -0.7421 \\ 0.0893 & -0.1250 & 0.0655 & -0.3304 & 0.2351 \\ 0.2321 & -0.1250 & -0.2520 & -0.2589 & 0.3224 \\ 0.1250 & -0.3750 & -0.0417 & -0.0625 & 0.3958 \\ -0.5000 & 0.5000 & 0.2778 & 0.7500 & -0.6389 \end{bmatrix}$$

Finalmente, se obtiene un resultado para $\mathbf{X} = \mathbf{Q_d A_S^{-1} Q_i B}$ como

$$\mathbf{X} = \begin{bmatrix} -0.2143 & 0.5000 & 0.1984 & 0.3929 & -0.7421 \\ 0.0893 & -0.1250 & 0.0655 & -0.3304 & 0.2351 \\ 0.2321 & -0.1250 & -0.2520 & -0.2589 & 0.3224 \\ 0.1250 & -0.3750 & -0.0417 & -0.0625 & 0.3958 \\ -0.5000 & 0.5000 & 0.2778 & 0.7500 & -0.6389 \end{bmatrix} \begin{bmatrix} 2 \\ 7 \\ 3 \\ 8 \\ 1 \end{bmatrix} = \begin{bmatrix} 6.0675 \\ -2.9077 \\ -2.9157 \\ -2.6042 \\ 8.6944 \end{bmatrix}$$

De aquí se obtiene la solución de un sistema de ecuaciones utilizando el método de factorización **QR**.

4.2.8 Matrices con formación especial (tipo banda)

Una forma de matriz que surge frecuentemente es la matriz tridiagonal, la cual tiene la forma [Burden *et al.*, 2002]

$$\begin{bmatrix} a_1 & c_1 & & & & & \\ b_2 & a_2 & c_2 & & & 0 & \\ & \cdots & \cdots & \cdots & & & \\ & & \cdots & \cdots & \cdots & & \\ & & & \cdots & \cdots & \cdots & \\ & 0 & & & b_{n-1} & a_{n-1} & c_{n-1} \\ & & & & & b_n & a_n \end{bmatrix} \tag{4.45}$$

Todos los elementos son cero, excepto aquellos que están sobre la diagonal principal y los que están arriba y abajo de ésta. Tales matrices se pueden encontrar en la solución de problemas de valor en la frontera para ecuaciones diferenciales ordinarias, las cuales satisfacen la condición $|a_i| \geq |b_i| + |c_i|$.

www.FreeLibros.me

Esta condición de matriz con un gran número de elementos cero, es particularmente apropiada para el uso de métodos iterativos. De hecho, debido a que los elementos están agrupados alrededor de la diagonal, es posible desarrollar una variante del método de eliminación gaussiana que sólo trabaje con los elementos diferentes a cero, y aproveche el hecho de que gran número de elementos lo son. Como el trabajo realizado para un método iterativo depende del número de iteraciones, en este caso es preferible el método directo.

Si la ecuación matricial es $\mathbf{AX} = \mathbf{V}$, entonces las ecuaciones para reducir a la forma triangular en este algoritmo, el cual se conoce como *algoritmo de Thomas*, son

$$\alpha_1 = a_1, \quad \gamma_1 = \frac{c_1}{\alpha_1}, \quad u_1 = \frac{v_1}{\alpha_1} \tag{4.46a}$$

$$\alpha_i = a_i - b_i \gamma_{i-1}, \quad (i = 2, 3, \ldots, n) \tag{4.46b}$$

$$u_i = \frac{(v_i - b_i u_{i-1})}{\alpha_i}, \quad (i = 2, 3, \ldots, n) \tag{4.47a}$$

$$\gamma_i = \frac{c_i}{\alpha_i}, \quad (i = 2, 3, \ldots, n-1) \tag{4.47b}$$

y la solución mediante la sustitución hacia atrás está dada por

$$x_n = u_n \tag{4.48a}$$

$$x_i = u_i - \gamma_i x_{i+1}, \quad (i = n-1, n-2, \ldots, 1) \tag{4.48b}$$

La ventaja de formular los cálculos de computación de esta manera es que, para ciertos conjuntos comunes de a_i, b_i y c_i, se puede demostrar que los multiplicadores γ_i tienen un módulo menor que la unidad, lo cual nos da el mismo efecto obtenido por pivoteo en la eliminación gaussiana ordinaria. El hecho descubierto en el algoritmo anterior es que el número de operaciones requeridas es, aproximadamente, igual a $5n$, que, para n grande, no se compara con el $\frac{n^3}{3}$ de la eliminación gaussiana sobre una matriz llena. Se debe observar también que se puede idear un esquema similar para matrices con elementos agrupados alrededor de la diagonal con ancho de la banda más grande que 3.

Otra forma de matriz común es la matriz de bloque tridiagonal que se genera en la solución en diferencias finitas de ecuaciones diferenciales parciales. La forma es similar a la ecuación (4.45), pero en este caso los elementos son matrices. Las ecuaciones (4.46), (4.47) y (4.48) se pueden utilizar para reducirla a una forma triangular, dado que la cantidad matricial α_i^{-1} remplaza la división entre α_i. El proceso utiliza el cálculo de una matriz inversa pero esto se puede hacer de manera más económica por eliminación gaussiana. Este método es considerablemente más eficiente que las iteraciones simples o la eliminación gaussiana total.

4.3 Métodos iterativos

Se sabe que es posible usar un esquema iterativo que mejore la solución hasta que se obtenga convergencia. Ésta es la esencia de este tipo de métodos empleados para resolver sistemas de ecuaciones lineales. Los más comunes son el método de Jacobi y el de Gauss-Seidel. Ambos parten del mismo principio, pero el esquema difiere en la forma de sustitución, como a continuación se describe.

4.3.1 Método de Jacobi

La forma más conveniente de expresar estos métodos es usar la notación matricial. La matriz $\mathbf{A}$ se divide en tres partes, correspondientes a los tres conjuntos de coeficientes. La ecuación $\mathbf{AX} = \mathbf{B}$ será

$$(\mathbf{L} + \mathbf{D} + \mathbf{U})\mathbf{X} = \mathbf{B} \tag{4.49}$$

www.FreeLibros.me

donde **L** es una matriz triangular inferior con ceros en la diagonal, **D** es la matriz diagonal que contiene los elementos de la diagonal de **A**, y **U** es una matriz triangular superior con ceros en la diagonal.

El método de Jacobi para la r-ésima iteración se escribe como sigue

$$\mathbf{DX}^{(r)} = -(\mathbf{L}+\mathbf{U})\mathbf{X}^{(r-1)} + \mathbf{B}, \qquad (r = 0, 1, \ldots) \tag{4.50}$$

El método de Jacobi converge para matrices **A** que son diagonalmente dominantes, en el sentido que es matemáticamente preciso, [Burden *et al.*, 2002], [Nieves *et al.*, 2002], [Rodríguez, 2003]. Para determinar si el método es convergente, si se tiene que la matriz de iteración

$$-\mathbf{D}^{-1}(\mathbf{L}+\mathbf{U}) \tag{4.51}$$

además de un término aditivo, $\mathbf{D}^{-1}\mathbf{B}$, mapea un conjunto de **X** en el siguiente. Esta matriz de iteración tiene valores propios, cada uno de los cuales refleja el factor por el que se suprime durante una iteración el valor propio particular de algún residual no deseado. Es evidente que estos factores tienen un módulo menor que 1. Con módulo menor que uno. La rapidez de convergencia de este método está dada por la rapidez de disminución del valor propio más lento: es decir, el factor con el módulo más grande. El módulo de este factor está, por consiguiente, entre 0 y 1, y se le llama *radio espectral del operador de relajación* (ρ_s). El número de iteraciones (r) necesarias para reducir el error total por un factor (10^{-p}) se estima por:

$$r \approx \frac{p\ln(10)}{-\ln(\rho_s)} \tag{4.52}$$

En el límite, cuando (ρ_s) tiende a 1, el sistema está en el punto de ruptura. Aquí puede o no converger, dependiendo de las condiciones iniciales. La idea de iterar se puede aplicar fácilmente a ecuaciones simultáneas lineales. Esto se ilustra mediante un ejemplo numérico. El código desarrollado en Matlab de este método se proporciona en la sección 4.6.10.

EJEMPLO 4.9

Mediante el método de Jacobi, resolver el conjunto de ecuaciones lineales dadas por:

$$\begin{aligned} 10x_1 + \quad x_2 + \quad x_3 &= 24 \\ -x_1 + 20x_2 + \quad x_3 &= 21 \\ x_1 - \; 2x_2 + 100x_3 &= 300 \end{aligned}$$

Despejando de la primera ecuación la primera variable y así sucesivamente, se obtiene

$$\begin{aligned} x_1 &= \frac{1}{10}(24 - x_2 - x_3) \\ x_2 &= \frac{1}{20}(21 + x_1 - x_3) \\ x_3 &= \frac{1}{100}(300 - x_1 + 2x_2) \end{aligned}$$

Si las primeras aproximaciones son $x_1 = 0$, $x_2 = 0$ y $x_3 = 0$ entonces la siguiente iteración es

$$x_1 = 2.4,\ x_2 = 1.05 \text{ y } x_3 = 3$$

y las iteraciones sucesivas son,

$$x_1 = 1.995,\ x_2 = 1.02 \text{ y } x_3 = 2.997$$

www.FreeLibros.me

y

$$x_1 = 1.9983,\ x_2 = 0.9999 \text{ y } x_3 = 2.9995$$

La sustitución de los valores $x_1 = 2,\ x_2 = 1$ y $x_3 = 3$ en el sistema inicial demuestra que el proceso está convergiendo rápidamente a la solución verdadera. Si se utiliza la ecuación (4.51) y la ecuación (4.52), se tiene que el radio espectral y el número de iteraciones para reducir el error total por un factor $\left(10^{-3}\right)$ son, respectivamente $(\rho_s = 0.0755)$ y $(r = 3)$.

Es claro que las iteraciones pueden ser un método muy útil de solución. Se demuestra ahora que éste también puede ser un método inapropiado en ciertas circunstancias. Las ecuaciones del ejemplo 4.9 se pueden tomar en un orden diferente, por ejemplo

$$\begin{aligned} x_1 - 2x_2 + 100x_3 &= 300 \\ 10x_1 + x_2 + x_3 &= 24 \\ -x_1 + 20x_2 + x_3 &= 21 \end{aligned}$$

Con este acomodo, el radio espectral del sistema es $(\rho_s = 27.7802)$, lo cual lo hace divergente. Si se calcula el número de iteraciones con la ecuación (4.52), se obtiene un número negativo, lo cual no tiene sentido.

A continuación se muestra que esta forma de ordenar las ecuaciones es divergente, resolviendo el sistema numéricamente. Despejando de la primera ecuación la primera variable y así sucesivamente, se obtiene

$$\begin{aligned} x_1 &= 300 + x_2 - 100x_3 \\ x_2 &= 24 - 10x_1 - x_3 \\ x_3 &= 21 + x_1 - 20x_2 \end{aligned}$$

Utilizando los valores $x_1 = 0,\ x_2 = 0$ y $x_3 = 0$ para la primera aproximación, se obtienen las aproximaciones sucesivas

$$x_1 = 300,\ x_2 = 24 \text{ y } x_3 = 21$$

y

$$x_1 = -1752,\ x_2 = -2997 \text{ y } x_3 = -3879$$

y no hay posibilidad de que esta secuencia converja a los valores $x_1 = 2,\ x_2 = 1$ y $x_3 = 3$. Por tanto, es crucial obtener algún conocimiento para que se pueda asegurar que las iteraciones converjan. El método anterior, conocido como *método de Jacobi*, rara vez se usa porque existen varias mejoras que se pueden utilizar para elevar la rapidez de convergencia. Sin embargo, si se cuenta con una computadora que efectúe cálculos en paralelo, se recomienda este método por ser altamente paralelizable.

4.3.2 Método de Gauss-Seidel

En el estudio del ejemplo anterior, se puede cuestionar por qué se encuentra un bloque de valores y después se sustituye siempre en las ecuaciones. Esto se ve razonable si los nuevos valores encontrados se sustituyen inmediatamente en las ecuaciones subsiguientes. Esta idea es la base del método de Gauss-Seidel que mejora en forma considerable la convergencia para ciertos tipos de ecuaciones [Burden *et al.*, 2002], [Nieves *et al.*, 2002], [Rodríguez, 2003]. Para ver la forma en que trabajan los diferentes métodos, es conveniente dividir los coeficientes en tres grupos que constituyen el llamado *conjunto de elementos diagonales*: los elementos en la diagonal, arriba de la diagonal y debajo de la diagonal.

Es conveniente escalar las ecuaciones dadas por la ecuación (4.49) mediante la división entre los elementos de la diagonal, de manera que **D** será igual a la matriz unitaria **I**. El método de Jacobi resulta de transferir todos los términos al lado derecho, excepto los términos de la diagonal, iterando como sigue

www.FreeLibros.me

$$\mathbf{X}^{(r+1)} = (-\mathbf{L} - \mathbf{U})\mathbf{X}^{(r)} + \mathbf{B}, \quad (r = 0, 1, \ldots) \tag{4.53}$$

Sin embargo, el método de Gauss-Seidel introduce $x_1^{(r+1)}$, $x_2^{(r+1)}$, etc., en el lado derecho de (4.53) tan pronto como éstos estén disponibles. Por tanto, las ecuaciones de iteración son

$$\mathbf{X}^{(r+1)} = -\mathbf{L}\mathbf{X}^{(r+1)} - \mathbf{U}\mathbf{X}^{(r)} + \mathbf{B} \tag{4.54a}$$

o

$$(\mathbf{I} + \mathbf{L})\mathbf{X}^{(r+1)} = -\mathbf{U}\mathbf{X}^{(r)} + \mathbf{B} \tag{4.54b}$$

Escribiendo estas ecuaciones en forma expandida:

$$\begin{aligned}
x_1^{(r+1)} &= -\left(a_{12}x_2^{(r)} + a_{13}x_3^{(r)} + a_{14}x_4^{(r)} + \cdots + a_{1n}x_n^{(r)}\right) + b_1 \\
x_2^{(r+1)} &= -\left(a_{21}x_1^{(r+1)} + a_{23}x_3^{(r)} + a_{24}x_4^{(r)} + \cdots + a_{2n}x_n^{(r)}\right) + b_2 \\
x_3^{(r+1)} &= -\left(a_{31}x_1^{(r+1)} + a_{32}x_2^{(r+1)} + a_{34}x_4^{(r)} + \cdots + a_{3n}x_n^{(r)}\right) + b_3 \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
x_n^{(r+1)} &= -\left(a_{n1}x_1^{(r+1)} + a_{n2}x_2^{(r+1)} + a_{n3}x_3^{(r+1)} + \cdots + a_{n(n-1)}x_{n-1}^{(r+1)}\right) + b_n
\end{aligned} \tag{4.55}$$

Cuando el método de Jacobi converge, se puede demostrar que el método de Gauss-Seidel converge más rápido, debido a que el radio espectral es justamente el cuadrado del radio espectral del método de Jacobi. En la sección 4.6.11 se proporciona el código desarrollado en Matlab para este método.

4.3.3 Sobrerrelajación

Si un proceso iterativo converge lentamente, algunas veces es posible tomar pasos más grandes para el cálculo de valores y, por tanto, acelerar la convergencia [Burden *et al.*, 2002], [Rodríguez, 2003]. Esta técnica se adopta casi siempre para ecuaciones lineales simultáneas que surgen de la solución de ecuaciones diferenciales elípticas. Estas ecuaciones satisfacen ciertas condiciones que garantizan la convergencia; pero la razón de convergencia, sobre todo para un sistema grande, es muy lenta. Por ello los valores calculados del proceso de Gauss-Seidel se ven como valores intermedios y se usa la siguiente ecuación para encontrar los valores modificados.

$$\mathbf{X}^{(r+1)} = \mathbf{X}^{(r)} + \omega\left(\tilde{\mathbf{X}}^{(r+1)} - \mathbf{X}^{(r)}\right) \tag{4.55}$$

aquí $\tilde{\mathbf{X}}^{(r+1)}$ es el valor calculado en el proceso de Gauss-Seidel. En la ecuación se puede observar que el nuevo valor se obtiene al multiplicar el incremento de la última aproximación por un factor ω. Cuando $\omega > 1$ tenemos el llamado *método de sobrerrelajación*. Por otro lado, si $\omega < 1$ el sistema de ecuaciones se llama *subrelajado*. Para ecuaciones diferenciales parciales elípticas, el valor ω generalmente satisface $1 < \omega < 2$. En términos de matrices, la ecuación sería:

$$\mathbf{X}^{(r+1)} = \mathbf{X}^{(r)} + \omega\left(-\mathbf{L}\mathbf{X}^{(r+1)} - \mathbf{U}\mathbf{X}^{(r)} + \mathbf{B} - \mathbf{X}^{(r)}\right) = [-\omega\mathbf{U} + (1-\omega)\mathbf{I}]\mathbf{X}^{(r)} - \omega\mathbf{L}\mathbf{X}^{(r+1)} + \omega\mathbf{B} \tag{4.56}$$

La elección de ω es la principal dificultad para usar el método de sobrerrelajación. Sin embargo, las matrices que surgen de la solución de ecuaciones diferenciales parciales tienen formas especialmente sencillas, y para algunas de ellas es posible calcular un valor preciso de ω. En particular, es preferible que surja un sobreestimado de ω que un subestimado. En el caso de una matriz más general, sería más razonable usar la subrelajación, si fuera posible conducir una serie de experimentos, con diferentes factores de relajación para determinar el valor más apropiado.

4.3.4 Convergencia de los métodos iterativos

Para investigar los diversos métodos, se escriben las ecuaciones anteriores en la forma general

$$\mathbf{X}^{(r+1)} = \mathbf{P}\mathbf{X}^{(r)} + \mathbf{C} \tag{4.57}$$

www.FreeLibros.me

La solución final **X** está dada por la ecuación

$$\mathbf{X} = \mathbf{PX} + \mathbf{C} \tag{4.58}$$

y, por tanto, el error $\mathbf{E}^{(r)} = \mathbf{X} - \mathbf{X}^{(r)}$ está dado por

$$\mathbf{X} - \mathbf{X}^{(r+1)} = \mathbf{P}\left(\mathbf{X} - \mathbf{X}^{(r)}\right) \tag{4.59}$$

es decir

$$\mathbf{E}^{(r+1)} = \mathbf{PE}^{(r)} = \mathbf{P}^{r+1}\mathbf{E}^{(0)} \tag{4.60a}$$

Está claro que la convergencia se puede obtener si el efecto de la matriz **P** es reducir el error $\mathbf{E}^{(r)}$. Una de las condiciones necesarias, es que todos los valores propios de **P** deben tener módulo menor que uno. Por otro lado, se tiene $|\lambda_i| > 1$ con un correspondiente vector propio $\mathbf{V}_i$. Entonces,

$$\mathbf{E}^{(0)} = \mathbf{V}_i \tag{4.61a}$$

y

$$\mathbf{E}^{(r+1)} = \mathbf{P}^{r+1}\mathbf{V}_i = \lambda^{r+1}\mathbf{V}_i \tag{4.61b}$$

el cual crece sin cota cuando r se incrementa. La condición $|\lambda_i| < 1$ también es una condición suficiente y, por tanto, si se conocen los valores propios de **P**, se determina cuándo convergen las iteraciones.

Los valores propios por sí mismos son difíciles de evaluar; pero hay varias condiciones que se pueden revisar para obtener una cota superior para su módulo. Si esta cota es menor que uno, entonces el proceso converge. Sin embargo, si la cota superior es más grande que la unidad, aún es posible tener los valores propios con módulos menores a la unidad, por lo que la condición de cota superior es suficiente, mas no es necesaria para la convergencia. En los dos primeros métodos iterativos considerados anteriormente, la matriz **P** tiene las formas

$$-\mathbf{D}^{-1}\left(\mathbf{L} + \mathbf{U}\right) \qquad \textit{Jacobi} \tag{4.62a}$$

$$-(\mathbf{I} + \mathbf{L})^{-1}\,\mathbf{U} \qquad \textit{Gauss-Seidel} \tag{4.62b}$$

Esto demuestra que la condición de la diagonal dominante en la matriz original **A** es suficiente para asegurar la convergencia del proceso representado por las matrices anteriores.

Una matriz se llama *diagonal estrictamente dominante* si:

$$d_r < 1 \text{ para } r = 1, 2, \ldots, n \tag{4.63}$$

donde

$$d_r = \frac{\sum_{j=1}^{n}{}' |a_{rj}|}{|a_{rr}|} \tag{4.64}$$

Con la notación prima se indica que el valor de a_{rr} se omite en la sumatoria. Si $d_r \le 1$ para $r = 1, 2, \ldots, n$ y $d_r < 1$ para al menos un valor de r, entonces la matriz se llama *diagonal débilmente dominante*. Esta condición es suficiente para la convergencia del proceso iterativo. Otra condición que puede garantizar la convergencia es cuando la matriz **A** es *definida positiva*. Debido a que esta propiedad es difícil de comprobar, se utiliza más la propiedad de diagonal dominante para verificar si la convergencia se puede garantizar.

4.3.5 Matrices dispersas

El término "dispersa" se utiliza para describir una matriz que tiene un gran número de elementos cero. Para este tipo de matrices, la magnitud del tiempo de cálculo para los esquemas directos e iterativos no

www.FreeLibros.me

sigue el modelo normal. En general, un método iterativo emplea mn^2 operaciones para su solución, donde m es el número de iteraciones y n es el número de ecuaciones. Sin embargo, si el método se programa para incluir los cálculos sólo para los elementos diferentes de cero, entonces la cantidad de cálculos es proporcional al número de elementos no nulos. Esto puede reducir el tiempo de cálculo a un nivel donde los métodos iterativos son más económicos que los métodos directos.

Hay dos situaciones donde éste no es exactamente el caso: Si la matriz no es diagonalmente dominante, entonces se debe utilizar un método directo para evitar problemas de convergencia, o si la matriz tiene una estructura especial, entonces se pueden diseñar algoritmos especiales para aprovechar esas propiedades.

En el caso de dispersión aleatoria, donde el número de elementos no nulos en la matriz es pequeño, 5 o 10%, hay algunas variaciones interesantes para el método estándar de eliminación gaussiana. El problema que surge en el método estándar es que cada resta de filas introduce otros elementos no nulos, y es posible rellenar muy rápidamente la matriz dispersa. Se han considerado varias estrategias para reducir la presencia de estos elementos extra no nulos en programas que son diseñados para utilizar sólo los elementos no nulos.

Una simple variante que ha tenido buenos resultados experimentales es usar una nueva estrategia de pivoteo. En cada etapa del proceso, la columna pivote seleccionada es aquella que tiene la mayor cantidad de elementos no nulos y un elemento satisfactorio, es decir mayor que uno, en la posición pivote. Es importante que el elemento en la posición pivote no sea muy pequeño, pues con la estrategia señalada aquí, el elemento pivote generalmente no va a ser el más grande, como lo fue en el método de eliminación gaussiana previo.

La otra excepción es cuando la matriz tiene una forma sencilla, la cual lleva a adoptar una forma más eficiente de eliminación gaussiana, por lo que el tiempo de cálculo no es más grande y es proporcional a $n^3/3$.

4.4 Casos especiales

Existen dos casos especiales en sistemas de ecuaciones lineales: el primero es cuando se tienen más incógnitas que ecuaciones. Este caso se menciona aquí como un sistema de ecuaciones subdeterminado. El segundo caso se refiere al hecho de tener más ecuaciones que incógnitas, es decir, un sistema de ecuaciones sobredeterminado. Ambos casos se describen a continuación.

4.4.1 Sistema de ecuaciones subdeterminado

Para el caso donde se tienen más incógnitas que ecuaciones, se dice que el número de soluciones es infinito, por ejemplo si se tiene el sistema de m ecuaciones y n incógnitas, donde $m<n$, dado por:

$$
\begin{aligned}
&a_{11}x_1 + a_{12}x_2 \ + \cdots + a_{1n}x_n = b_1 \\
&a_{21}x_1 + a_{22}x_2 \ + \cdots + a_{2n}x_n = b_2 \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
&a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m
\end{aligned}
\tag{4.65}
$$

El concepto de solución es la búsqueda de valores o funciones para las incógnitas que satisfacen la igualdad; es decir, que al sustituir la solución en la ecuación, se establece el paso de una igualdad a una equivalencia. Para el caso de un sistema de ecuaciones se debe cumplir en forma adicional el concepto de *simultaneidad;* es decir, que los valores o funciones encontrados satisfacen en forma simultánea todas las igualdades. Además, se tiene el concepto de solución única no nula, que se cumple cuando existe un solo grupo de valores, al menos uno de ellos diferente de cero, que satisfacen en forma simultánea un grupo de ecuaciones.

Para el caso específico del sistema de ecuaciones (4.65), se tiene un número infinito de soluciones. Si se reacomoda este sistema tomando $n=m+1$, se tiene que

www.FreeLibros.me

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 \ + \cdots + a_{1m}x_m &= b_1 - a_{1n}x_n \\
a_{21}x_1 + a_{22}x_2 \ + \cdots + a_{2m}x_m &= b_2 - a_{2n}x_n \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots&\cdots\cdots\cdots \\
a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mm}x_m &= b_m - a_{mn}x_n
\end{aligned}
\tag{4.66}
$$

Reacomodando en forma matricial, se obtiene

$$
\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mm} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \\ x_m \end{bmatrix} = \begin{bmatrix} b_1 - a_{1n}x_n \\ b_2 - a_{2n}x_n \\ \vdots \\ b_m - a_{mn}x_n \end{bmatrix} \tag{4.67}
$$

En notación compacta se tiene, entonces:

$$\mathbf{AX} = \mathbf{W} \tag{4.68}$$

donde $\mathbf{A}$ es una matriz cuadrada de $(m \times m)$, $\mathbf{X}$ y $\mathbf{W}$ son vectores de $(m \times 1)$. Si el determinante de $\mathbf{A}$ es diferente de cero, $\det(\mathbf{A}) \neq 0$, entonces el sistema (4.68) tiene solución única de la forma,

$$\mathbf{X} = \mathbf{A}^{-1}\mathbf{W} \tag{4.69}$$

Como no existe ninguna restricción sobre el valor que puede tomar x_n, entonces se tiene que $x_n \in (-\infty, +\infty)$; así se establece la relación funcional:

$$\mathbf{W} = f(x_n) \tag{4.70}$$

La ecuación (4.70) forma un número infinito de vectores. Por tanto, cada uno de ellos, sustituido en la ecuación (4.69) da una solución diferente para el vector $\mathbf{X}$ formado con $(x_1, x_2, \ldots, x_m)$. Así, se establece que el número de soluciones del sistema (4.66) es infinito; es decir, el número de conjuntos solución de un sistema subdeterminado es infinito. En la sección 4.6.12 se proporciona el código Matlab para resolver este tipo de sistemas.

4.4.2 Sistema de ecuaciones sobredeterminado

Para el caso donde se tienen más ecuaciones que incógnitas, si se supone que todas las ecuaciones son linealmente independientes, es decir, si el sistema es de rango pleno en renglones, si tiene exceso, se da el caso donde no se tiene un conjunto de valores o funciones que satisfagan el concepto de solución. En este caso lo único que se puede obtener es una aproximación, cuyo margen de precisión está determinado por la definición del error. Si se utiliza el concepto de error, éste se determina por el método de mínimos cuadrados. En el sistema de ecuaciones dado por

$$\mathbf{AX} = \mathbf{B} \tag{4.71}$$

donde $\mathbf{A}$ tiene más renglones que columnas, entonces la ecuación (4.71) se premultiplica por la transpuesta de $\mathbf{A}$, lo que equivale a tener el error que determina por el método de mínimos cuadrados, para obtener

$$\mathbf{A}^{\mathrm{T}}\mathbf{AX} = \mathbf{A}^{\mathrm{T}}\mathbf{B} \tag{4.72}$$

Si se definen nuevas variables, donde $\mathbf{D} = \mathbf{A}^{\mathrm{T}}\mathbf{A}$ es una matriz cuadrada y $\mathbf{F} = \mathbf{A}^{\mathrm{T}}\mathbf{B}$, se llega a:

$$\mathbf{DX} = \mathbf{F} \tag{4.73}$$

El sistema (4.73) tiene solución única si el determinante de $\mathbf{D}$ es diferente de cero. Esta solución es sólo una aproximación a la del sistema original definido por la ecuación (4.71). La solución que se obtiene por

www.FreeLibros.me

este método cumple con el concepto de mínimo *error cuadrado*, es decir *la suma mínima de los errores elevados al cuadrado*. Si se toma la ecuación (4.71) y se hace la igualación con el error, se tendrá entonces la definición del error

$$\mathbf{E} = \mathbf{AX} - \mathbf{B} \tag{4.74}$$

y el *error de la aproximación* dada por el método se define como *la suma de todos los errores elevados al cuadrado*, que es la función por minimizar, es decir, como

$$\sum_{i=1}^{n} (E_i)^2 \rightarrow \text{mín} \tag{4.76}$$

donde n es el número de ecuaciones. Para ejemplificar el caso anterior, la sección 4.6.13 proporciona el código desarrollado en Matlab.

4.5 Comparación de los métodos

El tiempo de cálculo necesario por cada método es una consideración importante. En la tabla 4.1 se dan detalles de las cantidades de operaciones para los diversos métodos. En muchas computadoras, los tiempos para la multiplicación y división son mayores que en la suma y la resta y, por tanto, la cantidad de operaciones es sólo para la multiplicación y la división.

Tabla 4.1 Cantidad de operaciones para los métodos estándar.

Método	Operaciones
Eliminación gaussiana	$\frac{n^3}{3} + n^2 - \frac{n}{3}$
Descomposición triangular	$\frac{n^3}{3} + n^2 - \frac{n}{3}$
Eliminación de Jordan	$\frac{n^3}{2} + n^2 - \frac{n}{2}$
Inversión de matriz y multiplicación	$n^3 + n^2$
Métodos iterativos	rn^2

n es el número de ecuaciones

r es el número de iteraciones necesarias para algún criterio específico de convergencia

En los métodos directos es claro que, aquellos que utilizan alguna estructura especial en la matriz ahorran operaciones en comparación con los métodos estándar. Para una matriz general que se va a resolver, se pueden utilizar los métodos directos como la eliminación gaussiana o la descomposición triangular. Si una computadora tiene la opción de acumular productos de doble precisión, entonces el algoritmo de descomposición triangular puede proporcionar mejores resultados. El método de descomposición triangular también es útil cuando la matriz es simétrica, ya que permite reducir el número de operaciones y recortar la necesidad de almacenar en un 50%.

Los métodos iterativos se utilizan para matrices con elementos dispersos donde la razón de convergencia es atractiva. Si el método converge lentamente por algún problema, sería preferible utilizar los métodos directos, aunque la matriz esté dispersa. El método de pivoteo que usa la estrategia de elegir en cada etapa siguiente la columna con el menor número de elementos no nulos, ha dado algunos resultados experimentales confiables; sin embargo, existe el peligro de que los errores puedan acumularse debido a la modificación del método de pivoteo normal.

En ausencia de cualquier otra guía, el método iterativo más utilizado es el método de Gauss-Seidel, ya que tiene una convergencia más acelerada que el método de Jacobi. Si la matriz tiene una forma especial,

www.FreeLibros.me

es decir, proviene de algunas ecuaciones diferenciales parciales, entonces es posible calcular un factor apropiado de relajación ω y utilizarlo para acelerar la convergencia. Para una matriz general, la elección de un factor de sobrerelajación es difícil; pero esto se puede valorar si la razón de convergencia es lenta y se van a realizar varios cálculos similares. Se pueden efectuar una serie de cálculos con valores experimentales de ω para así seleccionar un valor apropiado.

4.6 Programas desarrollados en Matlab

Esta sección proporciona los códigos de los programas desarrollados en Matlab para todos los ejercicios propuestos. A continuación se listan todos ellos:

4.6.1. Eliminación gaussiana
4.6.2. Eliminación de Gauss-Jordan
4.6.3. Inversa de una matriz
4.6.4. Inverso de una matriz con pivoteo parcial
4.6.5. Inverso de una matriz con pivoteo total
4.6.6. Factorización LU
4.6.7. Factorización Doolittle-Crout
4.6.8. Método de Choleski
4.6.9. Factorización QR
4.6.10. Método de Jacobi
4.6.11. Método de Gauss-Seidel
4.6.12. Sistema subdeterminado
4.6.13. Sistema sobredeterminado

4.6.1 Eliminación gaussiana

El método de eliminación gaussiana se basa en operaciones lineales entre ecuaciones; es decir, la multiplicación de un escalar por una ecuación, o la suma elemento a elemento de dos ecuaciones. Realizando estas dos operaciones de una forma ordenada, se llega a un sistema triangular superior, el cual se resuelve mediante una sustitución regresiva simple.

Programa principal del método de eliminación gaussiana

```
%  Programa para resolver un sistema de ecuaciones lineales utilizando la técnica de
%  eliminación gaussiana.
clear all
clc
% Matriz de coeficientes del sistema de ecuaciones lineales.
A = [ 5 4 8 1 6; 9 5 1 6 4; 4 1 9 8 6; 4 5 9 7 6; 6 2 9 7 1 ];
% Vector de entradas del sistema de ecuaciones.
B = [9; 1; 2; 9; 1];
% Número de incógnitas del sistema de ecuaciones.
N = rank(A);
% Proceso de eliminación gaussiana.
for k = 1:N-1
    for m = k+1:N
        MT = -A(m,k)/A(k,k);          % Multiplicadores.
        A(m,:) = A(m,:) + MT*A(k,:);   % Modificación de la matriz A.
        B(m)   = B(m)   + MT*B(k);     % Modificación del vector B.
    end
end
% Proceso de sustitución regresiva.
x(N) = B(N)/A(N,N);
for k = N-1:-1:1
    ind = N - k;
```

www.FreeLibros.me

```
    x(k) = 0;
    for m = 1:ind
        x(k) =  x(k) - A(k,k+m)*x(k+m);
    end
        x(k) = (B(k)+x(k))/A(k,k);
end
```

4.6.2 Eliminación de Gauss-Jordan

El método de eliminación Gauss-Jordan es una simple modificación del esquema de eliminación gaussiana. En este esquema se llega a una matriz diagonal, de tal forma que las soluciones quedan directamente en el vector de entradas.

Programa principal del método de eliminación de Gauss-Jordan

```
% Programa para resolver un sistema de ecuaciones lineales utilizando la técnica de
% eliminación de Gauss-Jordan.
clear all
clc
% Matriz de coeficientes del sistema de ecuaciones lineales.
A = [ 8 4 6 6 9; 5 3 2 6 3; 5 9 5 1 2; 9 3 4 4 9; 8 9 8 7 9 ];
% Vector de entradas del sistema de ecuaciones.
B = [3; 3; 3; 5; 7];
% Número de incógnitas del sistema de ecuaciones.
N = rank(A);
% Proceso de eliminación Gauss-Jordan.
for k = 1:N
        B(k)   = B(k)/A(k,k);    % Modificación del vector B debido a la
                                 % normalización del pivote.
        A(k,:) = A(k,:)/A(k,k);  % Normalización del pivote.
    for m = 1:N
        if m ~= k
            MT = - A(m,k);                 % Multiplicadores.
            A(m,:) = A(m,:) + MT*A(k,:);   % Modificación de la matriz A.
            B(m)   = B(m)   + MT*B(k);     % Modificación del vector B.
        end
    end
end  % NOTA: La solución queda en el vector B.
```

4.6.3 Inversa de una matriz

El método de la inversa es la aplicación de la eliminación Gauss-Jordan. La matriz por invertir emplea una matriz unitaria. El proceso concluye depositando la inversa en la matriz unitaria. De esta forma, la solución del sistema de ecuaciones se obtiene al multiplicar la inversa por el vector de valores.

Programa principal del método de la inversa

```
% Programa para resolver un sistema de ecuaciones lineales utilizando la técnica de
% la inversa de una matriz.
clear all
clc
% Matriz de coeficientes del sistema de ecuaciones lineales.
A = [ 9 2 9 2 5; 6 1 5 2 8; 6 7 8 3 7; 1 1 7 1 5; 9 7 2 3 5 ];
% Vector de entradas del sistema de ecuaciones.
B = [4; 1; 6; 1; 5];
% Número de incógnitas del sistema de ecuaciones.
N = rank(A);
% Matriz unitaria.
Id = diag(ones(N,1),0);
```

www.FreeLibros.me

```
% Proceso de eliminación Gauss-Jordan para determinar la inversa de una matriz.
for k = 1:N
        Id(k,:) = Id(k,:)/A(k,k);  % Modificación de la matriz Id debido a la
                                   % normalización del pivote.
        A(k,:) = A(k,:)/A(k,k);    % Normalización del pivote.
    for m = 1:N
        if m ~= k
            MT = - A(m,k);                    % Multiplicadores.
            A(m,:) = A(m,:) + MT*A(k,:);      % Modificación de la matriz A.
            Id(m,:) = Id(m,:) + MT*Id(k,:);   % Modificación de la matriz Id.
        end
    end
end
x = Id*B;   % NOTA: La inversa de la matriz queda en Id.
```

4.6.4 Inversa de una matriz con pivoteo parcial

El método de la inversa es la aplicación de la eliminación Gauss-Jordan. La matriz por invertir emplea una matriz unitaria. El proceso concluye depositando la inversa en la matriz unitaria. De esta forma, la solución del sistema de ecuaciones se obtiene al multiplicar la inversa por el vector de entradas. El caso de pivoteo parcial se refiere al hecho de que en cada caso se elige como elemento pivote el elemento de mayor módulo de la columna.

Programa principal del método de la inversa con pivoteo parcial

```
% Programa para resolver un sistema de ecuaciones lineales utilizando la técnica de
% la inversa de una matriz con pivoteo parcial, es decir, utilizando como pivote el
% elemento de la columna de mayor módulo.
clear all
clc
% Matriz de coeficientes del sistema de ecuaciones lineales.
A = [ 8 3 8 3 4; 4 3 3 2 6; 3 4 5 0 4; 3 1 3 1 1; 4 6 3 2 1 ];
% Vector de entradas del sistema de ecuaciones.
B = [3; 1; 3; 3; 0];
N = rank(A);    % Número de incógnitas del sistema de ecuaciones.
Id = eye(N);    % Matriz identidad.
A = [A Id];     % Se aumenta la matriz.
Ren = [1:N];    % Vector que numera el total de renglones.
PIV = [];       % Vector que acumula los renglones que van siendo pivotes.
% Proceso de eliminación Gauss-Jordan con pivoteo parcial para determinar la inversa
% de una matriz
for k = 1:N
    [Amp Pos] = max(abs(A(Ren,k)));  % Valor máximo de la k-ésima columna y su
                                     % posición.
    R = Ren(Pos);                    % Renglón correspondiente a la posición donde se
                                     % encuentra el pivote.
    PIV = [PIV R];                   % Acumula los renglones pivote.
    ROT(R,k)=1;                      % Matriz de rotación reacomodar las ecuaciones
                                     % en la posición adecuada.
    Ren = setdiff (Ren,PIV);         % Encuentra los valores diferentes entre Ren y
                                     % PIV para buscar el máximo sólo en las
                                     % posiciones que no han sido pivote.
    A(R,:) = A(R,:)/A(R,k);          % Normalización del pivote.
    for m = 1:N
        if m ~= R
            MT = - A(m,k);                 % Multiplicadores.
            A(m,:) = A(m,:) + MT*A(R,:);   % Modificación de la k-ésima columna de la
                                           % matriz A.
        end
    end
end
% Asignación de la matriz inversa contenida en la matriz aumentada.
```

www.FreeLibros.me

```
IA = A(1:N,N+1:2*N);
% Rotación de la matriz inversa para darle el acomodo correcto.
IA = ROT*IA;
% Cálculo de la solución.
x = IA*B;
```

4.6.5 Inversa de una matriz con pivoteo total

El método para obtener la inversa de una matriz mediante la aplicación la eliminación Gauss-Jordan, emplea una matriz unitaria. El proceso concluye depositando la inversa en la matriz unitaria. La solución del sistema de ecuaciones se logra al multiplicar la inversa por el vector de valores. El caso de pivoteo total se refiere al hecho de que en cada caso se elige como elemento pivote el elemento de mayor módulo de la matriz.

Programa principal del método de la inversa con pivoteo total

```
% Programa para resolver un sistema de ecuaciones lineales utilizando la técnica de
% la inversa de una matriz con pivoteo total, es decir, utilizando como pivote el
% elemento de mayor módulo.
clear all
clc
% Matriz de coeficientes del sistema de ecuaciones lineales.
A = [ 6   7   5   9   2
      8   1   7   0   1
      1   8   3   4   2
      7   1   7   1   5
      2   1   1   6   1 ];
% Vector de entradas del sistema de ecuaciones.
B = [7; 1; 6; 1; 4];
N = rank(A);    % Numero de incógnitas del sistema de ecuaciones.
Id = eye(N);    % Matriz identidad.
A = [A Id];     % Se aumenta la matriz.
Ren = [1:N];    % Vector que numera el total de renglones.
Col = [1:N];    % Vector que numera el total de columna.
PIR = [];       % Vector que acumula los renglones que van siendo pivotes.
PIC = [];       % Vector que acumula las columnas que van siendo pivotes.
% Proceso de eliminación Gauss-Jordan con pivoteo parcial para determinar la inversa
% de una matriz.
for k = 1:N
    [Amp PoR] = max(abs(A(Ren,Col)));   % Valor máximo de la k-ésima columna y su
                                        % posición.
    [AM PoC] = max(abs(Amp));           % Valor máximo de la k-ésima columna y su
                                        % posición.
    C = Col(PoC);                       % Renglón correspondiente a la posición donde
                                        % se encuentra el pivote.
    PIC = [PIC C];                      % Acumula las columnas pivote.
    R = Ren(PoR(PoC));                  % Renglón correspondiente a la posición donde
                                        % se encuentra el pivote.
    PIR = [PIR R];                      % Acumula los renglones pivote.
    ROT(R,C)=1;                         % Matriz de rotación reacomodar las
                                        % ecuaciones en la posición adecuada.
    Ren = setdiff(Ren,PIR);             % Encuentra los valores diferentes entre Ren
                                        % y PIR para buscar el máximo sólo en las
                                        % posiciones que no han sido pivote.
    Col = setdiff(Col,PIC);             % Encuentra los valores diferentes entre Col
                                        % y PIC para buscar el máximo sólo en las
                                        % posiciones que no han sido pivote.
    A(R,:) = A(R,:)/A(R,C);             % Normalización del pivote.
    for m = 1:N
        if m ~= R
            MT = - A(m,C);                  % Multiplicadores.
            A(m,:) = A(m,:) + MT*A(R,:);    % Modificación de la k-ésima columna de la
                                            % matriz A.
```

www.FreeLibros.me

```
        end
    end
end
IA = A(1:N,N+1:2*N);    % Asignación de la matriz inversa contenida en la matriz
                        % aumentada.
IA = (ROT.')*IA;        % Rotación de la matriz inversa para darle el acomodo correcto.
x = IA*B;               % Cálculo de la solución.
```

4.6.6 Factorización LU

El método de la factorización LU parte de la aplicación del proceso de eliminación gaussiana, seguido de un proceso algebraico para determinar los factores de una matriz cuadrada. La solución del sistema de ecuaciones se obtiene mediante un proceso con una sustitución regresiva, y otra progresiva.

Programa principal del método de factorización LU

```
% Programa para resolver un sistema de ecuaciones lineales utilizando la técnica de
% factorización LU.
clear all
clc
% Matriz de coeficientes del sistema de ecuaciones lineales.
A = [ 9 3 3 0 1; 4 4 5 5 8; 3 5 8 3 6; 8 4 4 6 1; 5 6 8 2 1 ];
% Vector de entradas del sistema de ecuaciones.
B = [4; 0; 6; 8; 5];
% Número de incógnitas del sistema de ecuaciones.
N = rank(A);
% Proceso de eliminación gaussiana para obtener U.
U = A;   % Inicialmente se pone la matriz A en una matriz U.
for k = 1:N-1
    for m = k+1:N
        MT = -U(m,k)/U(k,k);            % Multiplicadores.
        U(m,:) = U(m,:) + MT*U(k,:);    % Modificación de la matriz U.
    end
end
% Proceso para obtener L.
L = eye(N);                             % Se pone una matriz unitaria en L.
L(2:N,1) = A(2:N,1)/U(1,1);             % Se resuelve la primera columna.
for k = 2:N-1
    for m = k+1:N
        MT = 0;
        for l = 1:k-1
            MT = MT + L(m:N,l).*U(l,k);     % Se realiza la sumatoria de los
                                            % elementos conocidos.
        end
        L(m:N,k) = (A(m:N,k)-MT)./U(k,k);   % Se calcula la k-ésima columna de L.
    end
end
% Proceso de sustitución progresiva.
y(1) = B(1);   % Se calcula el primer elemento del vector auxiliar y
for k = 2:N
    ind = k-1;
    y(k) = 0;
    for m = 1:ind
        y(k) =  y(k) - L(k,k-m)*y(k-m);     % Se calcula la sumatoria de los
                                            % elementos conocidos.
    end
        y(k) = B(k)+y(k);                   % Se calcula el k-ésimo elemento
                                            % del vector auxiliar.
end
% Proceso de sustitución regresiva.
x(N) = y(N)/U(N,N);   % Se calcula la última variable.
for k = N-1:-1:1
```

www.FreeLibros.me

```
    ind = N - k;
    x(k) = 0;
    for m = 1:ind
        x(k) =  x(k) - U(k,k+m)*x(k+m);     % Se calcula la sumatoria de los
                                            % elementos conocidos.
    end
        x(k) = (y(k)+x(k))/U(k,k);          % Se calcula la k-ésima variable.
end
```

4.6.7 Factorización Doolittle-Crout

El método de la factorización Doolittle-Crout parte de aplicación del proceso de eliminación gaussiana, seguida de un proceso algebraico para determinar los factores de una matriz cuadrada. La solución al sistema de ecuaciones se logra mediante un proceso con una sustitución regresiva y otra progresiva. La única variante respecto a la factorización **LU** es que la matriz superior contiene unos en la diagonal.

Programa principal del método de factorización Doolittle-Crout

```
% Programa para resolver un sistema de ecuaciones lineales utilizando la técnica de
% factorización Doolittle-Crout.
clear all
clc
% Matriz de coeficientes del sistema de ecuaciones lineales.
A = [ 1 1 4 2 2; 2 5 3 6 6; 4 3 9 1 7; 6 5 2 7 1; 6 4 9 0 2 ];
% Vector de entradas del sistema de ecuaciones.
B = [5; 2; 7; 6; 6];
% Número de incógnitas del sistema de ecuaciones.
N = rank(A);
% Proceso de eliminación gaussiana para obtener la matriz triangular superior.
C = A;   % Inicialmente se pone la matriz A en una matriz C.
for k = 1:N
    C(k,:) = C(k,:)/C(k,k);            % Normalización del pivote.
    for m = k+1:N
        MT = -C(m,k)/C(k,k);           % Multiplicadores.
        C(m,:) = C(m,:) + MT*C(k,:);   % Modificación de la matriz C.
    end
end
% Proceso para obtener la matriz triangular inferior.
D = zeros(N);                          % Se pone una matriz de ceros en D.
D(1:N,1) = A(1:N,1)/C(1,1);            % Se resuelve la primera columna.
for k = 2:N
    for m = k:N
        MT = 0;
        for l = 1:k-1
            MT = MT + D(m:N,l).*C(l,k);     % Se realiza la sumatoria de los
                                            % elementos conocidos.
        end
        D(m:N,k) = (A(m:N,k)-MT)./C(k,k);   % Se calcula la k-ésima columna de D.
    end
end
% Proceso de sustitución progresiva.
y(1) = B(1)/D(1,1);   % Se calcula el primer elemento del vector auxiliar y
for k = 2:N
    ind = k-1;
    y(k) = 0;
    for m = 1:ind
        y(k) =  y(k) - D(k,k-m)*y(k-m);   % Se calcula la sumatoria de los elementos
                                          % conocidos.
    end
        y(k) = (B(k)+y(k))/D(k,k);        % Se calcula el k-ésimo elemento del vector
                                          % auxiliar.
end
```

www.FreeLibros.me

```
% Proceso de sustitución regresiva.
x(N) = y(N)/C(N,N);                          % Se calcula la última variable.
for k = N-1:-1:1
    ind = N - k;
    x(k) = 0;
    for m = 1:ind
        x(k) =  x(k) - C(k,k+m)*x(k+m);   % Se calcula la sumatoria de los elementos
                                          % conocidos.
    end
        x(k) = y(k)+x(k);                 % Se calcula la k-ésima variable.
end
```

4.6.8 Método de Cholesky

El método de Cholesky se aplica a matrices con la peculiaridad de tener simetría. El método como tal hace una factorización en dos matrices triangulares (inferior y superior) con la peculiaridad de que una es la transpuesta de la otra, por lo que sólo se necesita calcular una de ellas. La solución del sistema de ecuaciones se realiza mediante un proceso con una sustitución regresiva y otra progresiva.

Programa principal del método de Cholesky

```
% Programa para resolver un sistema de ecuaciones lineales utilizando la técnica de
% factorización de Cholesky.
clear all
clc
% Matriz de coeficientes del sistema de ecuaciones lineales.
A = [ 9 1 3 2 1; 1 7 1 3 4; 3 1 9 1 2; 2 3 1 7 1; 1 4 2 1 8 ];
% Vector de entradas del sistema de ecuaciones.
B = [2; 1; 4; 1; 3];
% Número de incógnitas del sistema de ecuaciones.
N = rank(A);
% Proceso para obtener la matriz triangular inferior.
CInf = zeros(N);
% Se calcula el primer pivote.
CInf(1,1)=sqrt(A(1,1));
% Se calcula la primera columna.
CInf(2:N,1)=A(2:N,1)./CInf(1,1);
% Se calculan los siguientes pivotes y las siguientes columnas:
for k=2:N
    CInf(k,k)=sqrt(A(k,k)-sum(CInf(k,1:k-1).^2));
    for m=k+1:N
        Suma = sum(CInf(m,1:k-1).*CInf(k,1:k-1));
        CInf(m,k)=(1/CInf(k,k))*(A(m,k)-Suma);
    end
end
% La matriz triangular superior es simplemente la transpuesta de la matriz triangular
% inferior.
CSup = CInf.';
% Proceso de sustitución progresiva.
y(1) = B(1)/CInf(1,1);   % Se calcula el primer elemento del vector auxiliar y
for k = 2:N
    ind = k-1;
    y(k) = 0;
    for m = 1:ind
        y(k) =  y(k) - CInf(k,k-m)*y(k-m);   % Se calcula la sumatoria de los
                                             % elementos conocidos.
    end
        y(k) = (B(k)+y(k))/CInf(k,k);        % Se calcula el k-ésimo elemento del
                                             % vector auxiliar.
end
% Proceso de sustitución regresiva.
x(N) = y(N)/CSup(N,N);   % Se calcula la última variable.
```

www.FreeLibros.me

```
for k = N-1:-1:1
    ind = N - k;
    x(k) = 0;
    for m = 1:ind
        x(k) =  x(k) - CSup(k,k+m)*x(k+m);    % Se calcula la sumatoria de los
                                              % elementos conocidos.
    end
        x(k) = (y(k)+x(k))/CSup(k,k);         % Se calcula la k-ésima variable.
end
```

4.6.9 Factorización QR

El método de factorización **QR** emplea la factorización repetida de una secuencia de matrices triangulares, con la particularidad de que se efectúan transformaciones ortogonales. La solución del sistema de ecuaciones utiliza la inversa de una matriz triangular superior y una multiplicación directa de tres matrices por el vector de valores.

Programa principal de la factorización QR

```
% Programa para resolver un sistema de ecuaciones lineales utilizando la técnica de
% la factorización Q-R.
clear all
clc
% Matriz de coeficientes del sistema de ecuaciones lineales.
Aqr = [ 9 5 8 9 1; 3 6 8 2 4; 5 7 3 6 2; 3 1 4 5 2; 1 5 6 3 4 ];
% Vector de entradas del sistema de ecuaciones.
B = [2; 3; 1; 3; 1];
N = rank(Aqr);    % Número de incógnitas del sistema de ecuaciones.
A1 = Aqr;         % Asignación de la matriz original a A1.
Qi = eye(N);      % Se inicializa Qi como matriz unitaria.
Qd = eye(N);      % Se inicializa Qd como matriz unitaria.
% Ciclo iterativo para calcular Qd y Qi para la triangulación.
for kr = 1:2:30
    [Q1,R1] = qr(A1);
    A2 = R1*Q1;
    [Q2,R2] = qr(A2);
    A1 = R2*Q2;
    % Actualización de Qd y Qi.
    Qd = Qd*Q1*Q2;
    Qi = inv(Q2)*inv(Q1)*Qi;
end
% Triangular superior a la que se llega utilizando el método QR.
As = Qi*Aqr*Qd;
% Solución del sistema de ecuaciones.
x = Qd*inv(As)*Qi*B;
```

4.6.10 Método de Jacobi

El método de Jacobi es un método iterativo. Aquí, de la primera ecuación se despeja la primera variable, y así sucesivamente. Para su implementación se necesitan condiciones iniciales, las cuales se sustituyen en forma simultánea en todas las ecuaciones. El proceso se detiene cuando se tienen dos resultados consecutivos que difieren en menos de una tolerancia especificada.

Programa principal del método de Jacobi

```
% Programa para resolver un sistema de ecuaciones lineales en forma iterativa por el
% método de Jacobi.
clear all
clc
```

www.FreeLibros.me

```
% Matriz de coeficientes del sistema de ecuaciones lineales.
A = [ 9 0 1 1 3; 2 8 2 1 0; 1 2 8 1 2; 2 3 1 7 1; 3 1 0 2 9 ];
% Vector de entradas del sistema de ecuaciones.
B = [7; 3; 8; 4; 6];
N = rank(A);     % Número de incógnitas del sistema de ecuaciones.
D = zeros(N);    % Inicializa la matriz D en ceros.
L = zeros(N);    % Inicializa la matriz L en ceros.
U = zeros(N);    % Inicializa la matriz U en ceros.
K1 = diag(A);    % Asigna la diagonal de A a un vector.
D = diag(K1);    % Asigna el vector con la diagonal de A a la diagonal de D.
A = A-D;         % Elimina la diagonal de la matriz A.
% Ciclo para hacer la asignación de las matrices triangulares.
for k = 1:N-1
    L(k:N,k) = A(k:N,k);
    U(k,k:N) = A(k,k:N);
end
% Cálculo del número de iteraciones.
F  = -inv(D)*(L+U);             % Cálculo de los valores propios.
Rs = max(abs(eig(F)));          % Módulo del valor propio de disminución más lento.
p  = 6;                         % Régimen de convergencia.
Ni = (p*log(10))/(-log(Rs));    % Número de iteraciones en valor real.
Nm = fix(Ni)+1;                 % Número de iteraciones enteras.
X  = zeros(N,Nm);               % Matriz de N * Nm donde se guardan los resultados de
                                % cada iteración.
X(:,1)=ones;                    % Condiciones iniciales para iniciar el proceso
                                % iterativo.
% Ciclo iterativo del método de Jacobi.
for k=2:Nm
    X(:,k) = F*X(:,k-1) + inv(D)*B;
end
X   % Muestra en la pantalla el resultado de todas las iteraciones.
```

4.6.11 Método de Gauss-Seidel

El método de Gauss-Seidel es un método iterativo. La única variante con respecto al método de Jacobi es que la sustitución de las variables es diferente. En este método, al encontrar una variable se sustituye inmediatamente en la ecuación siguiente. El proceso se detiene cuando se tienen dos resultados consecutivos que difieren en menos de una tolerancia especificada.

Programa principal del método de Gauss-Seidel

```
% Programa para resolver un sistema de ecuaciones lineales en forma iterativa por el
% método de Gauss-Seidel.
clear all
clc
% Matriz de coeficientes del sistema de ecuaciones lineales.
A = [ 9 0 2 3 1; 1 7 1 2 1; 1 1 8 2 1; 2 1 2 7 2; 1 1 2 1 9 ];
% Vector de entradas del sistema de ecuaciones.
B = [2; 6; 2; 6; 9];
% Número de incógnitas del sistema de ecuaciones.
N = rank(A);
% Ciclo iterativo para normalizar los pivotes de las ecuaciones.
for k = 1:N
    B(k,1)   = B(k,1)/A(k,k);
    A(k,:) = A(k,:)/A(k,k);
end
D = eye(N);      % La matriz D es igual a la matriz identidad.
L = zeros(N);    % Inicializa la matriz L en ceros.
U = zeros(N);    % Inicializa la matriz U en ceros.
A = A-D;         % Elimina la diagonal de la matriz A.
% Ciclo para hacer la asignación de las matrices triangulares.
for k = 1:N-1
    L(k:N,k) = A(k:N,k);
```

www.FreeLibros.me

```
    U(k,k:N) = A(k,k:N);
end
% Cálculo del número de iteraciones.
F  = -inv(D+L)*U;             % Cálculo de los valores propios.
Rs = max(abs(eig(F)));        % Módulo del valor propio de disminución más lento.
p  = 6;                       % Régimen de convergencia.
Ni = (p*log(10))/(-log(Rs));  % Número de iteraciones en valor real.
Nm = fix(Ni)+1;               % Número de iteraciones enteras.
X  = zeros(N,Nm);             % Matriz de N * Nm donde se guardan los resultados de
                              % cada iteración.
X(:,1)=ones;                  % Condiciones iniciales para iniciar el proceso
                              % iterativo.
% Ciclo iterativo del método de Gausss-Seidel.
for k=2:Nm
    X(:,k) = F*X(:,k-1) + inv(D+L)*B;
end
X   % Muestra en la pantalla el resultado de todas las iteraciones.
```

4.6.12 Sistema de ecuaciones subdeterminado

Cuando se tiene un sistema con más incógnitas que ecuaciones, a algunas incógnitas se les pueden asignar un valor aleatorio, para así resolver el resto del sistema de manera única. Si sólo hay una incógnita de más, entonces sólo se le asigna valor a una de las variables. La forma de solución es pasar en cada ecuación la incógnita a la cual se le asignó el valor a la parte derecha y así tener un sistema de n ecuaciones con n incógnitas con solución única.

Programa principal de un sistema de ecuaciones subdeterminado

```
% Programa para resolver un sistema de ecuaciones lineales cuando se tienen más
% incógnitas que ecuaciones.
clear all
clc
% Matriz de coeficientes del sistema de ecuaciones lineales.
A = [ 4 6 1 3 3 5; 2 4 3 3 6 2; 5 3 1 1 3 8; 6 4 2 2 8 3; 3 4 0 9 8 7 ];
% Vector de entradas del sistema de ecuaciones.
B = [9; 1; 2; 9; 1];
% Número de incógnitas del sistema de ecuaciones.
N = rank(A);
% Se pasa la última incógnita de cada ecuación al lado derecho, así se tiene un nuevo
% vector de valores. Para esto se le asigna un valor a esta variable. Como puede ser
% cualquier valor, las posibilidades son infinitas.
Xr = N/rand(1);         % Valor aleatorio de la última variable.
Bn = B - A(:,N+1)*Xr;   % Valor del nuevo vector de valores.
An = A(1:N,1:N);        % Asignación de la nueva matriz reducida en una variable.
Xn = inv(An)*Bn;        % Valor de las incógnitas restantes.
Xf = [Xn; Xr]           % Muestra en la pantalla el valor de todas las variables.
```

4.6.13 Sistema de ecuaciones sobredeterminado

Cuando se tiene un sistema con más ecuaciones que incógnitas, la forma de resolverlo tomando en cuenta todas las ecuaciones es la premultiplicación por la transpuesta de la matriz de datos. Esto equivale a utilizar el método de mínimos cuadrados. Con este proceso se llega a un sistema de n ecuaciones con n incógnitas cuya solución se puede implementar con el método de la matriz inversa.

Programa principal de un sistema de ecuaciones sobredeterminado

```
% Programa para resolver un sistema de ecuaciones lineales cuando se tienen más
% ecuaciones que incógnitas.
clear all
clc
```

www.FreeLibros.me

```
% Matriz de coeficientes del sistema de ecuaciones lineales.
A = [ 7   5   4; 5   3   6; 8   2   4; 5   7   1; 6   3   3 ];
% Vector de entradas del sistema de ecuaciones.
B = [3; 0; 5; 8; 4];
% Número de incógnitas del sistema de ecuaciones.
N = rank(A);
% La solución al sistema de ecuaciones se implementa en forma directa
X = inv(A.'*A)*A.'*B
```

Problemas propuestos

4.7.1 Por el método de eliminación gaussiana resuelva el siguiente sistema de ecuaciones lineales:

$$\begin{bmatrix} 1 & 7 & 5 & 9 \\ 4 & 2 & 7 & 1 \\ 9 & 6 & 8 & 2 \\ 3 & 7 & 3 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 6 \\ 9 \\ 4 \end{bmatrix}$$

4.7.2 Por el método de eliminación gaussiana resuelva el siguiente sistema de ecuaciones lineales:

$$\begin{bmatrix} 2 & 3 & 4 & 2 \\ 5 & 1 & 3 & 7 \\ 8 & 8 & 1 & 6 \\ 1 & 1 & 4 & 9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 4 \\ 8 \end{bmatrix}$$

4.7.3 Por el método de eliminación gaussiana resuelva el siguiente sistema de ecuaciones lineales:

$$\begin{bmatrix} 3 & 4 & 2 \\ 1 & 2 & 1 \\ 2 & 1 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 5 \\ 1 \end{bmatrix}$$

4.7.4 Por el método de eliminación gaussiana resuelva el siguiente sistema de ecuaciones lineales:

$$\begin{bmatrix} 1 & 2 & 8 \\ 7 & 7 & 5 \\ 9 & 3 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 8 \end{bmatrix}$$

4.7.5 Utilizando el método de Gauss-Jordan, resuelva el siguiente sistema de ecuaciones lineales:

$$\begin{bmatrix} 8 & 3 & 2 & 5 \\ 3 & 3 & 4 & 6 \\ 7 & 1 & 6 & 9 \\ 2 & 0 & 7 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 9 \\ 4 \\ 6 \end{bmatrix}$$

4.7.6 Utilizando el método de Gauss-Jordan, resuelva el siguiente sistema de ecuaciones lineales:

$$\begin{bmatrix} 1 & 1 & 5 & 2 \\ 4 & 4 & 3 & 1 \\ 0 & 6 & 6 & 2 \\ 5 & 7 & 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \\ 1 \\ 1 \end{bmatrix}$$

www.FreeLibros.me

4.7.7 Utilizando el método de Gauss-Jordan, resuelva el siguiente sistema de ecuaciones lineales:

$$\begin{bmatrix} 4 & 4 & 9 \\ 3 & 1 & 1 \\ 3 & 1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

4.7.8 Utilizando el método de Gauss-Jordan, calcule la inversa de la siguiente matriz **A**:

$$\mathbf{A} = \begin{bmatrix} 2 & 7 & 2 & 5 & 3 & 5 \\ 2 & 1 & 9 & 2 & 2 & 1 \\ 1 & 5 & 8 & 2 & 6 & 1 \\ 6 & 9 & 7 & 2 & 9 & 7 \\ 1 & 8 & 4 & 9 & 2 & 8 \\ 9 & 3 & 7 & 2 & 8 & 4 \end{bmatrix}$$

4.7.9 Utilizando el método de Gauss-Jordan, calcule la inversa de la siguiente matriz **A**:

$$\mathbf{A} = \begin{bmatrix} 3 & 2 & 6 & 5 \\ 6 & 5 & 5 & 3 \\ 2 & 8 & 9 & 3 \\ 7 & 9 & 3 & 3 \end{bmatrix}$$

4.7.10 Utilizando el método de Gauss-Jordan, calcule la inversa de la siguiente matriz **A**:

$$\mathbf{A} = \begin{bmatrix} 1 & 9 & 4 \\ 0 & 1 & 1 \\ 4 & 1 & 0 \end{bmatrix}$$

4.7.11 Por factorización **A** = **LU**, es decir triangular inferior y triangular superior, determine los factores de las matrices de **A**:

$$\mathbf{A} = \begin{bmatrix} 4 & 6 & 8 & 1 \\ 12 & 21 & 30 & 11 \\ 24 & 60 & 99 & 77 \\ 28 & 57 & 95 & 71 \end{bmatrix}$$

4.7.12 Por factorización **A** = **LU**, es decir triangular inferior y triangular superior, determine los factores de las matrices de **A**:

$$\mathbf{A} = \begin{bmatrix} 9 & 5 & 2 & 9 \\ 18 & 17 & 8 & 20 \\ 27 & 22 & 12 & 30 \\ 18 & 59 & 42 & 38 \end{bmatrix}$$

4.7.13 Por factorización **A** = **LU**, es decir triangular inferior y triangular superior, determine los factores de las matrices de **A**:

www.FreeLibros.me

$$\mathbf{A}=\begin{bmatrix}3 & 8 & 2 & 2\\ 21 & 58 & 15 & 21\\ 3 & 8 & 4 & 3\\ 6 & 26 & 27 & 49\end{bmatrix}$$

4.7.14 Por factorización $\mathbf{A}=\mathbf{LU}$, es decir triangular inferior y triangular superior, determine los factores de las matrices de **A**:

$$\mathbf{A}=\begin{bmatrix}0.6 & 1 & 2.2 & 4.2\\ 0.06 & 1 & 3.32 & 2.12\\ 1.08 & 3.24 & 10.12 & 11.58\\ 1.44 & 4.65 & 14.11 & 17.3\end{bmatrix}$$

4.7.15 Por factorización Doolittle-Crout $\mathbf{A}=\mathbf{DC}$, es decir triangular inferior y triangular superior, determine los factores de las matrices de **A**:

$$\mathbf{A}=\begin{bmatrix}4 & 16 & 28 & 36\\ 5 & 22 & 39 & 51\\ 8 & 33 & 59 & 80\\ 7 & 31 & 57 & 91\end{bmatrix}$$

4.7.16 Por factorización Doolittle-Crout $\mathbf{A}=\mathbf{DC}$, es decir triangular inferior y triangular superior, determine los factores de las matrices de **A**:

$$\mathbf{A}=\begin{bmatrix}7 & 14 & 35\\ 1 & 11 & 68\\ 2 & 5 & 21\end{bmatrix}$$

4.7.17 Por factorización Doolittle-Crout $\mathbf{A}=\mathbf{DC}$, es decir triangular inferior y triangular superior, determine los factores de las matrices de **A**:

$$\mathbf{A}=\begin{bmatrix}7 & 14 & 21 & 7\\ 3 & 14 & 81 & 51\\ 9 & 21 & 55 & 30\\ 1 & 5 & 35 & 41\end{bmatrix}$$

4.7.18 Por factorización Doolittle-Crout $\mathbf{A}=\mathbf{DC}$, es decir triangular inferior y triangular superior, determine los factores de las matrices de **A**:

$$\mathbf{A}=\begin{bmatrix}3.4 & 14.96 & 10.88 & 6.12\\ 2.3 & 12.52 & 12.64 & 10.62\\ 3.2 & 15.48 & 17.32 & 15.94\\ 1.2 & 8.78 & 15.74 & 24.33\end{bmatrix}$$

4.7.19 Por el método de Cholesky, factorice la siguiente matriz **A**:

www.FreeLibros.me

$$\mathbf{A}=\begin{bmatrix}9 & 18 & 9 & 6\\ 18 & 40 & 20 & 18\\ 9 & 20 & 11 & 11\\ 6 & 18 & 11 & 42\end{bmatrix}$$

4.7.20 Por el método de Cholesky, factorice la siguiente matriz **A**:

$$\mathbf{A}=\begin{bmatrix}29 & 5 & 8 & 9 & 1\\ 5 & 61 & 8 & 12 & 4\\ 8 & 8 & 93 & 6 & 32\\ 9 & 12 & 6 & 65 & 2\\ 1 & 4 & 32 & 2 & 24\end{bmatrix}$$

4.7.21 Por el método de Cholesky, factorice la siguiente matriz **A**:

$$\mathbf{A}=\begin{bmatrix}28 & 4 & 1 & 7 & 1\\ 4 & 32 & 2 & 5 & 1\\ 1 & 2 & 75 & 9 & 8\\ 7 & 5 & 9 & 65 & 3\\ 1 & 1 & 8 & 3 & 19\end{bmatrix}$$

4.7.22 Por el método de Cholesky, factorice la siguiente matriz **A**:

$$\mathbf{A}=\begin{bmatrix}89 & 11 & 8 & 22 & 13 & 7\\ 11 & 59 & 12 & 7 & 23 & 1\\ 8 & 12 & 82 & 19 & 2 & 22\\ 22 & 7 & 19 & 74 & 8 & 4\\ 13 & 23 & 2 & 8 & 99 & 47\\ 7 & 1 & 22 & 4 & 47 & 79\end{bmatrix}$$

4.7.23 Utilizando la factorización $\mathbf{A}=\mathbf{QR}$, resuelva el sistema de ecuaciones $\mathbf{AX}=\mathbf{B}$ con los siguientes datos:

$$\begin{bmatrix}2 & 6 & 1 & 1 & 0\\ 5 & 8 & 1 & 3 & 2\\ 2 & 1 & 6 & 7 & 1\\ 4 & 5 & 9 & 6 & 2\\ 6 & 9 & 1 & 4 & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\\ x_5\end{bmatrix}=\begin{bmatrix}5\\ 5\\ 2\\ 1\\ 9\end{bmatrix}$$

4.7.24 Utilizando la factorización $\mathbf{A}=\mathbf{QR}$, resuelva el sistema de ecuaciones $\mathbf{AX}=\mathbf{B}$ con los siguientes datos:

$$\begin{bmatrix}5 & 1 & 6 & 2\\ 8 & 2 & 4 & 5\\ 7 & 1 & 1 & 7\\ 9 & 4 & 3 & 7\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\end{bmatrix}=\begin{bmatrix}7\\ 4\\ 1\\ 5\end{bmatrix}$$

www.FreeLibros.me

4.7.25 Utilizando la factorización $\mathbf{A}=\mathbf{QR}$, resuelva el sistema de ecuaciones $\mathbf{AX}=\mathbf{B}$ con los siguientes datos:

$$\begin{bmatrix} 1 & 6 & 7 & 2 \\ 9 & 5 & 8 & 4 \\ 4 & 3 & 5 & 1 \\ 7 & 6 & 1 & 9 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}=\begin{bmatrix} 1 \\ 3 \\ 9 \\ 2 \end{bmatrix}$$

4.7.26 Utilizando la factorización $\mathbf{A}=\mathbf{QR}$, resuelva el sistema de ecuaciones $\mathbf{AX}=\mathbf{B}$ con los siguientes datos:

$$\begin{bmatrix} 8 & 1 & 4 & 3 \\ 1 & 9 & 4 & 2 \\ 3 & 2 & 7 & 1 \\ 1 & 1 & 2 & 8 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}=\begin{bmatrix} 3 \\ 1 \\ 3 \\ 7 \end{bmatrix}$$

4.7.27 Por el método de Jacobi y el de Gauss-Seidel, resuelva el siguiente sistema de ecuaciones. De la primera ecuación despeje la primer variable y así sucesivamente. Calcule el radio espectral de cada método y el número de iteraciones para tener un error global de (10^{-3}). Las condiciones iniciales para ambos métodos son $x_1=0$, $x_2=0$ y $x_3=0$.

$$\begin{aligned} 4x_1-4x_2+2x_3&=1 \\ 2x_1+8x_2-2x_3&=2 \\ 3x_1-2x_2+3x_3&=1 \end{aligned}$$

Haga una tabla de las iteraciones y verifique cuál de los dos métodos converge más rápido a la solución exacta. Obtenga la solución exacta para efectos de comparación utilizando cualquier método directo, como la inversa.

4.7.28 Por el método de Jacobi y el de Gauss-Seidel, resuelva el siguiente sistema de ecuaciones. De la primera ecuación despeje la primer variable y así sucesivamente. Calcule el radio espectral de cada método y el número de iteraciones para tener un error global de (10^{-5}). Las condiciones iniciales para ambos métodos son $x_1=0$, $x_2=0$, $x_3=0$ y $x_4=0$.

$$\begin{aligned} 6x_1-2x_2+3x_3+5x_4&=6 \\ 5x_1+10x_2-6x_3+x_4&=8 \\ 3x_1+x_2+8x_3-3x_4&=3 \\ x_1+x_2+x_3-8x_4&=1 \end{aligned}$$

Haga una tabla de las iteraciones y verifique cuál de los dos métodos converge más rápido a la solución exacta. Obtenga la solución exacta para efectos de comparación utilizando cualquier método directo, como la inversa.

4.7.29 Para el siguiente sistema de ecuaciones subdeterminado, encuentre la solución si se tiene que $x_4=1$.

$$\begin{aligned} 5x_1+9x_2+2x_3+7x_4&=4 \\ 1x_1+5x_2+8x_3+3x_4&=5 \\ 4x_1+7x_2+2x_3+8x_4&=2 \end{aligned}$$

www.FreeLibros.me

4.7.30 Para el siguiente sistema de ecuaciones subdeterminado, encuentre la solución si se tiene que $x_4 = 3$.

$$\begin{aligned} 5x_1 + 9x_2 + 2x_3 + 8x_4 + 3x_5 &= 9 \\ 3x_1 + 7x_2 + 6x_3 + 4x_4 + 2x_5 &= 6 \\ 9x_1 + 3x_2 + 4x_3 + 6x_4 + 7x_5 &= 7 \\ 4x_1 + 3x_2 + 7x_3 + 6x_4 + 1x_5 &= 8 \end{aligned}$$

4.7.31 Para el siguiente sistema de ecuaciones subdeterminado, encuentre la solución si se tiene que $x_5 = 3$ y $x_4 = 2$.

$$\begin{aligned} 4x_1 + 2x_2 + 1x_3 + 8x_4 + 6x_5 &= 2 \\ 3x_1 + 4x_2 + 2x_3 + 1x_4 + 3x_4 &= 4 \\ 3x_1 + 8x_2 + 1x_3 + 2x_4 + 4x_4 &= 1 \end{aligned}$$

4.7.32 Para el siguiente sistema de ecuaciones subdeterminado, encuentre la solución si se tiene que $x_7 = 5$ y $x_6 = 1$.

$$\begin{aligned} 9x_1 + 3x_2 + 4x_3 + 6x_4 + 2x_5 + 7x_6 + 6x_7 &= 6 \\ 1x_1 + 2x_2 + 8x_3 + 3x_4 + 4x_5 + 6x_6 + 7x_7 &= 8 \\ 9x_1 + 3x_2 + 5x_3 + 7x_4 + 6x_5 + 4x_6 + 2x_7 &= 5 \\ 2x_1 + 4x_2 + 8x_3 + 6x_4 + 1x_5 + 9x_6 + 3x_7 &= 8 \\ 8x_1 + 5x_2 + 2x_3 + 6x_4 + 4x_5 + 7x_6 + 4x_7 &= 4 \end{aligned}$$

4.7.33 Encuentre la solución de un sistema sobredeterminado si se tienen siete ecuaciones y sólo cuatro incógnitas.

$$\begin{aligned} 3x_1 + 2x_2 + 1x_3 + 2x_4 &= 2 \\ 5x_1 + 6x_2 + 4x_3 + 5x_4 &= 7 \\ 7x_1 + 9x_2 + 2x_3 + 8x_4 &= 1 \\ 9x_1 + 2x_2 + 9x_3 + 2x_4 &= 5 \\ 2x_1 + 1x_2 + 3x_3 + 1x_4 &= 2 \\ 5x_1 + 9x_2 + 7x_3 + 3x_4 &= 7 \\ 4x_1 + 3x_2 + 1x_3 + 2x_4 &= 3 \end{aligned}$$

4.7.34 Encuentre la solución de un sistema sobredeterminado si se tienen seis ecuaciones y sólo dos incógnitas.

$$\begin{aligned} 7x_1 + 3x_2 &= 6 \\ 6x_1 + 1x_2 &= 8 \\ 2x_1 + 1x_2 &= 5 \\ 1x_1 + 6x_2 &= 6 \\ 6x_1 + 2x_2 &= 4 \\ 9x_1 + 1x_2 &= 2 \end{aligned}$$

4.7.35 Encuentre la solución de un sistema sobredeterminado si se tienen ocho ecuaciones y sólo dos incógnitas.

www.FreeLibros.me

$$1x_1 + 4x_2 = 1$$
$$6x_1 + 6x_2 = 2$$
$$11x_1 + 8x_2 = 3$$
$$14x_1 + 10x_2 = 4$$
$$18x_1 + 12x_2 = 5$$
$$21x_1 + 15x_2 = 6$$
$$26x_1 + 16x_2 = 7$$
$$31x_1 + 17x_2 = 8$$

4.7.36 Encuentre la solución de un sistema sobredeterminado si se tienen seis ecuaciones y sólo tres incógnitas.

$$9x_1 + 7x_2 + 6x_3 = 4$$
$$5x_1 + 9x_2 + 5x_3 = 3$$
$$9x_1 + 3x_2 + 6x_3 = 2$$
$$7x_1 + 5x_2 + 7x_3 = 4$$
$$4x_1 + 5x_2 + 5x_3 = 8$$
$$8x_1 + 7x_2 + 9x_3 = 6$$
$$7x_1 + 3x_2 + 8x_3 = 6$$
$$3x_1 + 5x_2 + 3x_3 = 5$$

www.FreeLibros.me

www.FreeLibros.me

Capítulo 5

Interpolación y ajuste de curvas

5.1 Aproximación e interpolación

Existen dos tipos de problemas que se pueden considerar dentro de este concepto; primero, el problema de interpolación que involucra el encontrar valores intermedios cuando los valores están dados como un conjunto finito de datos y, segundo, el problema de aproximarse a una función dentro de un intervalo, por medio de una función simple, por ejemplo un polinomio. Claramente, la función aproximada debe ser capaz de reducir el error de aproximación tanto como sea posible; por supuesto, diferentes formas de definir el error corresponden a diferentes métodos. En el caso de interpolación por el método de diferencias finitas, la función aproximada es un polinomio que tiene valores iguales a los dados en un conjunto finito de puntos. Denotando con $f_i \ (i = 0, 1, ..., n)$ los valores de $f(x)$ en los puntos $x_i \ (i = 0, 1, ..., n)$ y $\phi_n(x)$ la función aproximada; si se define el error como:

$$E_n = \sum_{i=0}^{n} \left| \phi_n(x_i) - f_i \right| \tag{5.1}$$

entonces ajustando la función exacta en los $n+1$ puntos, el error E_n se reduce a cero. Cierto que el error definido por la ecuación (5.1) se ha minimizado, pero la pregunta que falta por resolver es si los valores en los puntos donde $x \neq x_i$ tienen una buena aproximación, ya que en los problemas de aproximación se considera el error en todos los puntos dentro del intervalo.

Cuando se considera el error sobre un intervalo completo, el objetivo final es hacer el máximo error tan pequeño como sea posible. Éste es el tipo de aproximación minimizada donde el error se define como:

$$E_{\text{máx}} = \max_{a \leq x \leq b} \left| \phi(x) - f(x) \right| \tag{5.2}$$

y la función $\phi(x)$ se escoge de tal forma que $E_{\text{máx}}$ se minimice. Es en este contexto donde los polinomios de Tchebyshev han encontrado amplias aplicaciones.

El tercer caso de interés es cuando el número de datos de los cuales se tiene el valor es considerablemente más grande que el grado de la aproximación deseada. Es decir, si se quiere usar un polinomio de bajo orden, por ejemplo un polinomio cúbico, para una aproximación sobre un intervalo en el cual tal vez se conocen veinte valores de una función. Cuatro puntos son suficientes para determinar un polinomio cúbico único, y el

www.FreeLibros.me

error será entonces elevado en los puntos restantes. En esta situación, más que hacer el error cero en un punto en particular, se requiere que el error sea tan pequeño como sea posible sobre todo el intervalo. Una elección apropiada para definir el error en este caso está dado por:

$$S_m = \sum_{i=0}^{m} [\phi_n(x_i) - f_i]^2, \quad m \geq n \tag{5.3}$$

Así, el ajuste de mínimos cuadrados se obtiene encontrando la función $\phi_n(x)$ que minimiza el valor de S_m. El subíndice n implica que la función $\phi_n(x)$ depende de un número de parámetros a los cuales se les puede dar el valor apropiado para obtener el ajuste deseado. En el caso de un polinomio, estos parámetros son los coeficientes $a_0, a_1, \ldots, a_n$ y, por tanto, la función $\phi_n(x)$ tendrá $n+1$ parámetros variables.

Una gran ventaja de los polinomios es que usando operaciones aritméticas accesibles en una computadora digital, es posible evaluarlos directamente o hacer el cociente de dos polinomios. También es muy fácil evaluar integrales y derivadas de polinomios por cálculo directo. En cambio, el cálculo de otras funciones, como exponenciales o trigonométricas, se hace con métodos de aproximación.

Es importante conocer qué tan precisa se puede obtener una aproximación usando polinomios. Afortunadamente el teorema de Weierstrass afirma que, para cualquier función continua dentro de un intervalo finito, el error mínimo se puede hacer tan pequeño como se quiera eligiendo un polinomio de orden suficientemente alto. Otro tipo de aproximación, que también es significativa, es la aproximación por series de Fourier. En este caso se puede mostrar que, arbitrariamente, se puede obtener una buena aproximación para muchas clases de funciones, si éstas satisfacen las condiciones de Dirichlet, lo cual se enuncia en el siguiente teorema:

Teorema 5.1 Si $f(t)$ es una función periódica acotada en todo el periodo, tiene un número finito de máximos y mínimos locales, y tiene un número finito de puntos de discontinuidad, entonces la serie de Fourier de $f(t)$ converge a $f(t)$ en todos los puntos en los que $f(t)$ es continua y, al promedio de los límites por la derecha y por la izquierda de $f(t)$ en los puntos de discontinuidad.

Las condiciones del teorema 5.1, conocidas normalmente como **condiciones de Dirichlet**, aclaran que no es necesario que una función sea continua para tener un desarrollo válido en series de Fourier. Esto significa que una función se puede definir por intervalos mediante funciones totalmente diferentes. De esta forma, un mismo punto está evaluado por la izquierda por una función y por la derecha por otra, haciendo de éste un punto de discontinuidad. Aun así, si el número de puntos de discontinuidad es finito, se tiene una representación válida mediante la serie de Fourier.

5.2 Interpolación

Definiendo el **problema de interpolación** como: Dados un conjunto de $n+1$ puntos $\{(x_0, y_0), \ldots, (x_n, y_n)\}$ con $x_i \neq x_j, i \neq j$ determinar un polinomio de grado menor o igual que n, $P_n(x)$, tal que $P_n(x_i) = y_i$; $i = 0, 1, \ldots, n$, este problema se puede abordar considerando el polinomio

$$P_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n, \text{ con } P_n(x_i) = y_i; \; i = 0, 1, \ldots, n.$$

Se tiene entonces que

$$\begin{aligned} a_0 + a_1 x_0 + a_2 x_0^2 + \cdots + a_n x_0^n &= y_0 \\ a_0 + a_1 x_1 + a_2 x_1^2 + \cdots + a_n x_1^n &= y_1 \\ &\vdots \\ a_0 + a_1 x_n + a_2 x_n^2 + \cdots + a_n x_n^n &= y_n \end{aligned}$$

www.FreeLibros.me

Éste es un sistema con $n+1$ incógnitas y $n+1$ ecuaciones, el cual se puede escribir como

$$\mathbf{VA}=\mathbf{Y} \tag{5.4}$$

donde

$$\mathbf{V}=\begin{bmatrix} 1 & x_0 & x_0^2 & & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ 1 & x_2 & x_x^2 & & x_2^n \\ & \vdots & & \ddots & \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{bmatrix}$$

$$\mathbf{A}=[a_0\cdots a_n]^T$$

y

$$\mathbf{Y}=[y_0\cdots y_n]^T$$

La matriz **V** se llama **matriz de Vandermonde** y se tiene que el $\det \mathbf{V}\neq 0$ si y sólo sí $x_i\neq x_j$, $i\neq j$, por lo que el sistema tiene solución única. El programa desarrollado en Matlab se proporciona en la sección 5.9.1. Con esto queda demostrado el siguiente teorema.

Teorema 5.2 Dados los puntos $(x_0,y_0),\ldots,(x_n,y_n)$ con $x_i\neq x_j$, $i\neq j$ entonces existe un único polinomio $P_n(x)$, de grado menor o igual que n tal que $P_n(x_i)=y_i$; $i=0,1,\ldots,n$.

La construcción del polinomio interpolador mediante la solución del sistema (5.4) no es la adecuada, ya que la matriz de Vandermonde cuando se obtiene de esta forma por lo general magnifica los errores de redondeo de la solución del sistema. A este tipo de matrices se les conoce como matrices mal condicionadas. Para evitar la solución del sistema (5.4) se propone la formulación de Lagrange para determinar el error de aproximar una función mediante un polinomio interpolador. Se tiene el siguiente teorema:

Teorema 5.3 Sea $a=x_0<x_1<x_2<\cdots<x_n=b$, $f\in C^{(n+1)}[a,b]$ (función continua de orden $n+1$) y $f(x_i)=y_i$. Si $P(x)$ es un polinomio interpolador de f en los puntos $(x_i,\ y_i)$, $i=0,1,\ldots,n$, entonces

$$f(x)=P(x)+(x-x_0)(x-x_1)\cdots(x-x_n)\frac{f^{n+1}(\xi)}{(n+1)!} \tag{5.5}$$

donde $\xi\in(a,b)\,\forall x\in[a,b]$ y $\xi=\xi(x)$.

Demostración Para demostrar este resultado se hace uso del **teorema de Rolle (generalizado),** el cual establece que dada una función $g\in C^{(n)}[a,b]$ y si $a=x_0<x_1<x_2<\ldots<x_n=b$ con $g(x_0)=g(x_1)=\cdots=g(x_n)$, entonces existe $\xi\in(a,b)$ tal que $g^{(n)}(\xi)=0$.

Si $x=x_i$, $i=0,1,\ldots,n$ la relación se cumple. Suponiendo que $x\neq x_i$ y definiendo

$$g(t)=[f(t)-P(t)]-[f(x)-P(x)]\frac{(t-x_0)(t-x_1)\cdots(t-x_n)}{(x-x_0)(x-x_1)\cdots(x-x_n)}$$

entonces $g\in C^{(n+1)}[a,b]$. Ahora, $g(x)=0$, $g(x_i)=0$, $i=0,1,\ldots,n$ entonces g tienen al menos $(n+2)$ raíces en $[a,b]$.

www.FreeLibros.me

Aplicando el teorema de Rolle generalizado, existe $\xi \in (a,b)$ tal que $g^{(n+1)}(\xi)=0$. Esto es,

$$\frac{d^{(n+1)}}{dt^{(n+1)}}g(t)=f^{(n+1)}(t)-P^{(n+1)}(t)-\frac{f(x)-P(x)}{\prod_{i=0}^{n}(x-x_i)}\frac{d^{(n+1)}}{dt^{(n+1)}}\prod_{i=0}^{n}(t-x_i)$$

dado que

$$P^{(n+1)}(t)=0$$

y

$$\frac{d^{(n+1)}}{dt^{(n+1)}}\prod_{i=0}^{n}(t-x_i)=(n+1)!$$

entonces

$$\frac{d^{(n+1)}}{dt^{(n+1)}}g(\xi)=f^{(n+1)}(\xi)-\frac{f(x)-P(x)}{\prod_{i=0}^{n}(x-x_i)}(n+1)!=0$$

Despejando $f(x)-P(x)$, queda demostrado el teorema donde

$$a=x_0<x_1<x_2<\cdots<x_n=b$$

$$e_n=f(x)-P(x)=(x-x_0)(x-x_1)\cdots(x-x_n)\frac{f^{(n+1)}(\xi)}{(n+1)!}$$

Sabiendo que si $f\in C^{(n+1)}[a,b]$, existe $K>0$ tal que

$$\left|f^{(n+1)}(x)\right|<K \;\; \forall \;\; x\in[a,b]$$

por lo que

$$|e_n(x)|\le\frac{K}{(n+1)!}\prod_{i=0}^{n}|x-x_i| \qquad \textbf{(5.6)}$$

Este último resultado establece que el error permanecerá acotado a todo lo largo del intervalo de interpolación; por ejemplo:

EJEMPLO 5.1

Sea $f(x)=x^3, x\in[-1,1]$. El polinomio interpolador de primer orden a f en los puntos $(-1,-1)$, $(0, 0)$ y $(1, 1)$ es $p(x)=x$. Las gráficas de $f(x)$ y $p(x)$ están dadas en la figura 5.1.

El error de interpolación es

$$e(x)=x^3-x$$

Si se tiene que

$$e'(x)=3x^2-1$$

www.FreeLibros.me

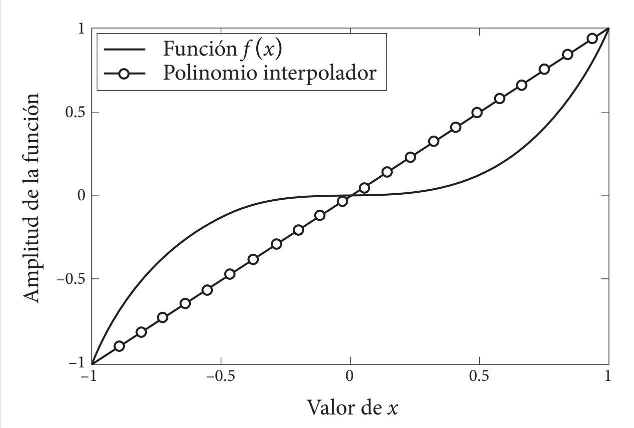


Figura 5.1 Gráfica de la función $f(x)$ y del polinomio interpolador $p(x)$.

los máximos y mínimos se alcanzan cuando $e'(x)=0$ por lo que $x=\pm\frac{1}{\sqrt{3}}\approx\pm 0.57735$. Entonces $e\left(\pm\frac{1}{\sqrt{3}}\right)=\mp 0.3849$. Por tanto, el **error absoluto** máximo es de 0.3849, el cual es mucho menor que la cota para el error obtenida con (5.6) que es:

$$|e(x)|\leq 3$$

Considerando otros puntos de interpolación, es decir los puntos $x_0=-\frac{\sqrt{3}}{2}$, $x_1=0$ y $x_2=\frac{\sqrt{3}}{2}$, se tiene que el polinomio interpolador de primer orden es $p(x)=\frac{3}{4}x$. Las gráficas de la función y el nuevo polinomio interpolador se muestran en la figura 5.2.

El error en este caso está definido por:

$$e(x)=f(x)-P(x)=x^3-\tfrac{3}{4}x$$

Además, de la fórmula (5.6), se sigue que

$$|e_n(x)|\leq 3\left|\left(x+\tfrac{\sqrt{3}}{2}\right)\left(x-\tfrac{\sqrt{3}}{2}\right)\right|=3\left|x^2-\tfrac{3}{4}\right|\leq\tfrac{9}{4}=2.25$$

Ahora

$$e(x)=f(x)-P(x)=x^3-\tfrac{3}{4}x$$

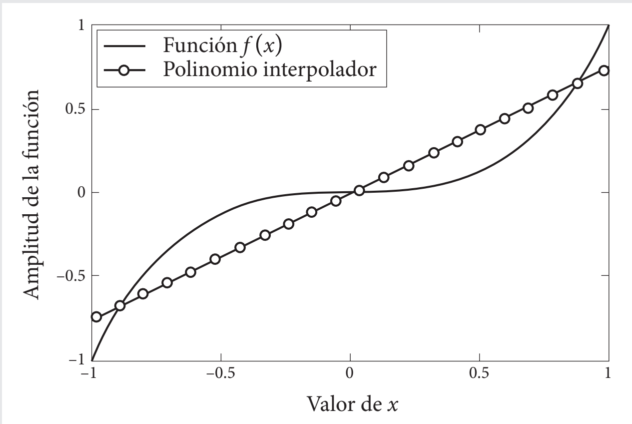


Figura 5.2 Gráfica de la función $f(x)$ y del polinomio interpolador $p(x)$.

www.FreeLibros.me

Entonces el error máximo se tiene cuando $e'(x)=0$; esto es, $x=\pm\frac{1}{2}$, y el error máximo es $e\left(\pm\frac{1}{2}\right)=\mp 0.25$. Así,

$$|e(x)|\leq 0.25 \quad \forall x\in[-1,1]$$

De estos resultados se puede observar que la elección de los puntos de interpolación es importante a fin de reducir el error de interpolación. Esto se trabajará con los polinomios de Tchebyshev.

5.2.1 Interpolación de Lagrange

Definiendo ahora los **polinomios fundamentales de Lagrange** como [Nakamura, 1992], [Maron, 1995], [Mathews, 2000], [Burden *et al.*, 2002], [Nieves *et al.*, 2002], [Rodríguez, 2003]:

$$L_{n,i}(x)=\prod_{\substack{j=0\\ j\neq i}}^{n}\frac{x-x_j}{x_i-x_j}=\frac{(x-x_0)(x-x_1)\cdots(x-x_{i-1})(x-x_{i+1})\cdots(x-x_n)}{(x_i-x_0)(x_i-x_1)\cdots(x_i-x_{i-1})(x_i-x_{i+1})\cdots(x_i-x_n)},\ i=0,1,\ldots,n$$

Se tiene que

$$L_{n,i}(x_i)=1,\ L_{n,i}(x_j)=0,\ i\neq j$$

y $L_i(x)$ es un polinomio de grado n. Definiendo además

$$P_n(x)=y_0L_{n,0}(x)+y_1L_{n,1}(x)+\cdots+y_nL_{n,n}(x)=\sum_{i=0}^{n}y_iL_{n,i}(x) \tag{5.7}$$

se tiene

$$\begin{aligned}P_n(x_0)&=y_0\\ P_n(x_1)&=y_1\\ &\vdots\\ P_n(x_n)&=y_n\end{aligned}$$

De esto se sigue que P_n es el polinomio interpolador de grado menor o igual a n. La fórmula (5.5) se puede escribir con la ayuda de (5.7) como

$$f(x)=\sum_{i=0}^{n}y_iL_{n,i}(x)+(x-x_0)(x-x_1)\cdots(x-x_n)\frac{f^{(n+1)}(\xi)}{(n+1)!}$$

donde $y_i=f(x_i)$. Si se requiere evaluar el polinomio interpolador en algún valor dado de x, es conveniente utilizar la expresión (5.7) directamente. La sección 5.9.2 provee el código en Matlab para la construcción de un interpolador de Lagrange de cualquier orden, dependiendo de la cantidad de datos que se tengan. El siguiente ejemplo considera ese caso.

EJEMPLO 5.2

Construir un polinomio interpolador utilizando el método de Lagrange considerando los puntos (1, 3), (2, 4), (3, 2) y (5, 1). En este caso $n=3$; entonces se tiene que

$$L_{3,0}(x)=\frac{(x-2)(x-3)(x-5)}{(1-2)(1-3)(1-5)}=-\frac{1}{8}(x-2)(x-3)(x-5)$$

$$L_{3,1}(x)=\frac{(x-1)(x-3)(x-5)}{(2-1)(2-3)(2-5)}=\frac{1}{3}(x-1)(x-3)(x-5)$$

www.FreeLibros.me

$$L_{3,2}(x)=\frac{(x-1)(x-2)(x-5)}{(3-1)(3-2)(3-5)}=-\frac{1}{4}(x-1)(x-2)(x-5)$$

$$L_{3,3}(x)=\frac{(x-1)(x-2)(x-3)}{(5-1)(5-2)(5-3)}=\frac{1}{24}(x-1)(x-2)(x-3)$$

Se tiene entonces

$$P_3(x)=-\frac{3}{8}(x-2)(x-3)(x-5)+\frac{4}{3}(x-1)(x-3)(x-5)$$
$$-\frac{1}{2}(x-1)(x-2)(x-5)+\frac{1}{24}(x-1)(x-2)(x-3)$$

Simplificando,

$$P_3(x)=\frac{1}{2}x^3-\frac{9}{2}x^2+11x-4$$

La gráfica del polinomio está dada en la figura 5.3.

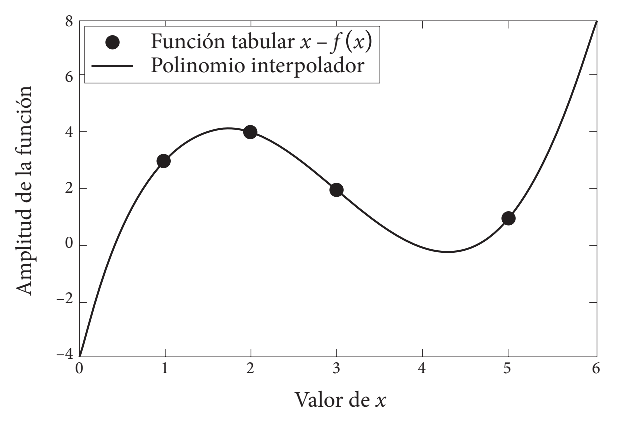


Figura 5.3 Gráfica de la función tabular y del polinomio interpolador $P_3(x)$.

Si se requiere evaluar el polinomio interpolador en sólo un punto, por ejemplo $x=4$, simplemente se considera

$$L_{3,0}(4)=\frac{(4-2)(4-3)(4-5)}{(1-2)(1-3)(1-5)}=\frac{-2}{-8}=\frac{1}{4}$$

$$L_{3,1}(4)=\frac{(4-1)(4-3)(4-5)}{(2-1)(2-3)(2-5)}=\frac{-3}{3}=-1$$

$$L_{3,2}(4)=\frac{(4-1)(4-2)(4-5)}{(3-1)(3-2)(3-5)}=\frac{-6}{-4}=\frac{3}{2}$$

$$L_{3,3}(4)=\frac{(4-1)(4-2)(4-3)}{(5-1)(5-2)(5-3)}=\frac{6}{24}=\frac{1}{4}$$

Por lo que

$$P(x=4)=3\left(\frac{1}{4}\right)+4(-1)+2\left(\frac{3}{2}\right)+1\left(\frac{1}{4}\right)=0$$

Esta formulación tiene la desventaja de que, si se desea agregar un punto extra al conjunto de puntos, se deben volver a realizar todos los cálculos para la obtención del polinomio. En la siguiente sección se establecerá una nueva formulación para la cual es relativamente fácil agregar un nuevo dato al conjunto de puntos de interpolación y aprovechar los cálculos efectuados.

www.FreeLibros.me

5.2.2 Formulación de Newton con diferencias divididas

Se sabe que, si se tiene un polinomio de grado máximo n que satisface la condición

$$P_n(x_i) = y_i \qquad i = 0, 1, \ldots, n,$$

entonces este polinomio es único si todos los puntos x_i son distintos. De esta forma, el problema es reacomodar el polinomio en forma más conveniente para el cálculo automático. En forma particular, puede ser posible agregar puntos adicionales en forma simple sin invalidar los cálculos previos. Suponiendo que se tiene un polinomio interpolador $P_k(x)$ de máximo grado k que se ajusta a los datos en los puntos $x_i (i = 0, 1, \ldots, k)$ y se desea formar el término siguiente $P_{k+1}(x)$ agregando un punto de interpolación adicional x_{k+1}, como se requiere que el cálculo previo no se vea alterado, se busca la forma

$$P_{k+1}(x) = P_k(x) + q_{k+1}(x) \qquad k = 0, 1, \ldots, n-1 \tag{5.8}$$

donde $q_{k+1}(x)$ tiene un máximo grado de $k+1$. Debido a que $P_{k+1}(x)$ y $P_k(x)$ interpolan en los puntos $x_i (i = 0, 1, \ldots, k)$, se tiene

$$P_{k+1}(x_i) = P_k(x_i) \qquad i = 0, 1, \ldots, k \tag{5.9}$$

y por tanto, usando la ecuación (5.8) se tiene $q_{k+1}(x_i) = 0$. La elección adecuada para $q_{k+1}(x)$ es por tanto:

$$q_{k+1}(x) = a_{k+1}(x - x_0)(x - x_1)\cdots(x - x_k)$$

El polinomio interpolador resultante es conocido como polinomio interpolador de diferencias divididas de Newton [Maron, 1995], [Mathews, 2000], [Nieves *et al.*, 2002], [Rodríguez, 2003], y tiene la siguiente forma:

$$P_n(x) = a_0 + a_1(x - x_0) + \cdots + a_n(x - x_0)(x - x_1)\cdots(x - x_{n-1}) \tag{5.10}$$

Esto se puede expresar en forma recursiva, lo cual es conveniente para su cálculo. Así se tiene:

$$P_3(x) = \{[a_3(x - x_2) + a_2](x - x_1) + a_1\}(x - x_0) + a_0$$

Afortunadamente, los coeficientes a_k se pueden generar simplemente construyendo una tabla de diferencias divididas. Sustituyendo los valores de $x_i (i = 0, 1, \ldots, n-1)$ en la ecuación (5.10) y utilizando $P(x_i) = y_i$, se obtiene

$$\begin{aligned}
y_0 &= a_0 \\
y_1 &= a_0 + a_1(x_1 - x_0) \\
y_2 &= a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1) \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots
\end{aligned}$$

Los coeficientes a_i están dados por

$$a_0 = y_0 \qquad\qquad a_1 = \frac{y_1 - y_0}{x_1 - x_0}$$

$$a_2 = \frac{\dfrac{(y_2 - y_1)}{(x_2 - x_1)} - \dfrac{(y_1 - y_0)}{(x_1 - x_0)}}{(x_2 - x_0)} \qquad\qquad a_3 = \frac{\dfrac{\frac{(y_3 - y_2)}{(x_3 - x_2)} - \frac{(y_2 - y_1)}{(x_2 - x_1)}}{(x_3 - x_1)} - \dfrac{\frac{(y_2 - y_1)}{(x_2 - x_1)} - \frac{(y_1 - y_0)}{(x_1 - x_0)}}{(x_2 - x_0)}}{(x_3 - x_0)}$$

Para simplificar la notación, se definen las diferencias divididas como

0-ésima diferencia dividida $\qquad f[x_i] = y_i$

www.FreeLibros.me

1ra. diferencia dividida $f[x_i, x_{i+1}] = \dfrac{y_{i+1} - y_i}{x_{i+1} - x_i}$

$\vdots$

***k*-ésima diferencia dividida** $f[x_i, x_{i+1}, \ldots, x_{i+k}] = (f[x_i, x_{i+1}, \ldots, x_{i+k-1}] - f[x_{i+1}, \ldots, x_{i+k}])/(x_{i+k} - x_i)$

con lo que resulta que

$$
\begin{aligned}
a_0 &= f[x_0] \\
a_1 &= f[x_0, x_1] \\
a_2 &= f[x_0, x_1, x_2] \\
&\vdots \\
a_n &= f[x_0, x_1, \ldots, x_n]
\end{aligned}
$$

y se tiene que el polinomio interpolador está dado por

$$
\begin{aligned}
P(x) &= f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\
&\quad + \cdots + f[x_0, x_1, \ldots, x_n](x - x_0)(x - x_1)\cdots(x - x_{n-1}) \\
&= \sum_{k=0}^{n} f[x_0, x_1, \ldots, x_k](x - x_0)(x - x_1)\cdots(x - x_{k-1})
\end{aligned} \tag{5.11}
$$

Esta formulación del polinomio interpolador recibe el nombre de formulación de Newton con diferencias divididas. Se puede demostrar que

$$f[x_0, x_1, \ldots, x_n, x] = \frac{f^{(n+1)}(\varsigma)}{(n+1)!}$$

donde ς denota algún punto entre los puntos de interpolación para obtener de (5.5) la fórmula

$$f(x) = \sum_{k=0}^{n} f[x_0, x_1, \ldots, x_k](x - x_0)(x - x_1)\cdots(x - x_{k-1}) + (x - x_0)(x - x_1)\cdots(x - x_n) f[x_0, x_1, \ldots, x_n, x]$$

Los coeficientes son calculados construyendo una tabla de diferencias divididas (tabla 5.1), como se muestra a continuación. Entonces se tiene una forma de acomodo en la cual los coeficientes son fáciles de evaluar. La forma final nos da un calculo rápido para muchos valores de x con un algoritmo de multiplicación recursiva. Se pueden agregar puntos de interpolación adicionales para darle más precisión al esquema. La tabla de diferencias divididas muestra que el agregar un punto adicional simplemente introduce otra línea diagonal en la parte baja de la tabla, dando un coeficiente adicional a_{k+1} el cual no altera el conjunto anterior.

Tabla 5.1 Tabla de diferencias divididas de Newton.

		1a. diferencia	2a. diferencia	3a. diferencia
x_0	f_0			
		$\dfrac{f_1 - f_0}{x_1 - x_0} \equiv f_{01}$		
x_1	f_1		$\dfrac{f_{12} - f_{01}}{x_2 - x_0} \equiv f_{012}$	
		$\dfrac{f_2 - f_1}{x_2 - x_1} \equiv f_{12}$		$\dfrac{f_{123} - f_{012}}{x_3 - x_0} \equiv f_{0123}$

(*continúa*)

www.FreeLibros.me

		1a. diferencia	2a. diferencia	3a. diferencia
x_2	f_2		$\frac{f_{23}-f_{12}}{x_3-x_1} \equiv f_{123}$	
		$\frac{f_3-f_2}{x_3-x_2} \equiv f_{23}$		
x_3	f_3			

El código Matlab para implementar este método se proporciona en la sección 5.9.3 de este capítulo. El programa como tal determina los coeficientes de un polinomio interpolador utilizando la técnica de diferencias divididas de Newton.

EJEMPLO 5.3

Para construir el polinomio interpolador que pasa por los puntos (1, –6), (2, –67), (4, –495), (7, 2598) y (–2, –147), construir una tabla de diferencias divididas

Tabla 5.2 Tabla de diferencias divididas de Newton para el ejemplo 5.3.

x_i	Diferencia dividida				
	$f(x_i) = 1a.$	$2a.$	$3a.$	$4a.$	$5a.$
1	–6				
		–61			
2	–67		–51		
		–214		50	
4	–495		249		6
		1031		32	
7	2598		121		
		305			
–2	–147				

El polinomio interpolador está dado por

$$P_4(x) = -6 - 61(x-1) - 51(x-1)(x-2) + 50(x-1)(x-2)(x-4) + 6(x-1)(x-2)(x-4)(x-7)$$

Simplificando, se obtiene finalmente

$$P_4(x) = 6x^4 - 34x^3 - 23x^2 + 156x - 111$$

5.2.3 Formulación de Newton para puntos igualmente espaciados

Considerando ahora que los $n+1$ puntos de interpolación están igualmente espaciados por una distancia $h = \frac{b-a}{n}$ en el intervalo $[a, b]$, esto es

$$x_i = a + ih, \quad i = 0, 1, \ldots, n,$$

www.FreeLibros.me

o bien

$$x_i = x_0 + ih, \quad i = 0, 1, \ldots, n$$

sea $y_i = f(x_i)$, $i = 0, 1, \ldots, n$. Las diferencias divididas satisfacen

$$\begin{aligned}
f[x_i] &= y_i \\
f[x_i, x_{i+1}] &= \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \\
&= \frac{1}{h}(y_{i+1} - y_i) \\
f[x_i, x_{i+1}, x_{i+2}] &= \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i} \\
&= \frac{\frac{1}{h}(y_{i+2} - y_{i+1}) - \frac{1}{h}(y_{i+1} - y_i)}{2h} \\
&= \frac{1}{2h^2}(y_{i+2} - 2y_{i+1} + y_i) \\
f[x_i, x_{i+1}, x_{i+2}, x_{i+3}] &= \frac{f[x_{i+1}, x_{i+2}, x_{i+3}] - f[x_i, x_{i+1}, x_{i+2}]}{x_{i+3} - x_i} \\
&= \frac{\frac{1}{2h^2}(y_{i+3} - 2y_{i+2} + y_{i+1}) - \frac{1}{2h^2}(y_{i+2} - 2y_{i+1} + y_i)}{3h} \\
&= \frac{1}{3!h^3}(y_{i+3} - 3y_{i+2} + 3y_{i+1} - y_i)
\end{aligned}$$

También

$$f[x_i, x_{i+1}, x_{i+2}, x_{i+3}, x_{i+4}] = \frac{1}{4!h^4}(y_{i+4} - 4y_{i+3} + 6y_{i+2} - 4y_{i+1} + y_i)$$

Para simplificar la notación se definen las diferencias adelantadas como

$$\begin{aligned}
\Delta^0 y_i &= y_i \\
\Delta^1 y_i &= \Delta y_i = y_{i+1} - y_i \\
\Delta^k y_i &= \Delta(\Delta^{k-1} y_i) \text{ en forma lineal}
\end{aligned}$$

Entonces se tiene que

$$\begin{aligned}
\Delta^0 y_0 &= y_0 \\
\Delta^1 y_0 &= y_1 - y_0 \\
\Delta^2 y_0 &= \Delta(\Delta y_0) = \Delta(y_1 - y_0) \\
&= \Delta y_1 - \Delta y_0 = (y_2 - y_1) - (y_1 - y_0) \\
&= y_2 - 2y_1 + y_0 \\
\Delta^3 y_0 &= \Delta(\Delta^2 y_0) = \Delta(y_2 - 2y_1 + y_0) \\
&= \Delta y_2 - 2\Delta y_1 + \Delta y_0 = (y_3 - y_2) - 2(y_2 - y_1) + (y_1 - y_0) \\
&= y_3 - 3y_2 + 3y_1 - y_0
\end{aligned}$$

Una fórmula general para la k-ésima diferencia es

www.FreeLibros.me

$$\Delta^k y_0 = \sum_{j=0}^{k} (-1)^j \binom{k}{j} y_{k-j}$$

Usando esta notación se tiene que

$$f[x_0, x_1, \ldots, x_k] = \frac{1}{k!h^k} \Delta^k y_0$$

Si se considera el cambio de variable

$$x = x_0 + sh$$

se tiene entonces que

$$\begin{aligned} x - x_k &= (x_0 + sh) - (x_0 + kh) \\ &= (s - k)h \end{aligned}$$

y por tanto,

$$\begin{aligned} (x - x_0)(x - x_1)\cdots(x - x_{k-1}) &= (sh)((s-1)h)\cdots((s-k+1)h) \\ &= h^k s(s-1)\cdots(s-k+1) \end{aligned}$$

Definiendo los coeficientes binomiales generalizados como

$$\binom{s}{k} = \frac{s(s-1)\cdots(s-k+1)}{k!}, \text{ donde "}s\text{" y "}k\text{" son números reales,}$$

se tiene que éstos son generalización de los coeficientes binomiales para s y k enteros y

$$(x - x_0)(x - x_1)\cdots(x - x_{k-1}) = h^k k! \binom{s}{k}$$

Con estas notaciones, se tiene que el polinomio interpolador de Newton (5.11) para puntos igualmente espaciados se puede escribir como

$$P(x) = \sum_{k=0}^{n} \frac{1}{k!h^k} \Delta^k y_0 h^k k! \binom{s}{k} = \sum_{k=0}^{n} \Delta^k y_0 \binom{s}{k}, \; x = x_0 + sh$$

Esta formulación del polinomio interpolador se llama polinomio interpolador de Newton con diferencias adelantadas para puntos igualmente espaciados. De forma similar a como se definieron las diferencias adelantadas, es posible definir las diferencias atrasadas por medio de

$$\begin{aligned} \nabla^0 y_i &= y_i \\ \nabla^1 y_i &= \nabla y_i = y_i - y_{i-1} \\ \nabla^k y_i &= \nabla\left(\nabla^{k-1} y_i\right) \text{ en forma lineal} \end{aligned}$$

Si se tiene que

$$\begin{aligned} \nabla^0 y_i &= y_i \\ \nabla^1 y_i &= y_i - y_{i-1} \\ \nabla^2 y_i &= \nabla(\nabla y_i) = \nabla(y_i - y_{i-1}) \\ &= \nabla y_i - \nabla y_{i-1} = (y_i - y_{i-1}) - (y_{i-1} - y_{i-2}) \\ &= y_i - 2y_{i-1} + y_{i-2} \end{aligned}$$

www.FreeLibros.me

una fórmula general para la k-ésima diferencia atrasada es

$$\nabla^k y_i = \sum_{j=0}^{k} (-1)^j \binom{k}{j} y_{i-j}$$

Usando esta notación, el polinomio interpolador se puede escribir como

$$p(x) = \sum_{k=0}^{n} \nabla^k y_n (-1)^k \binom{-s}{k}, \quad x = x_n + sh$$

Para el cómputo del factor $(-1)^k \binom{-s}{k}$, se tiene que

$$\begin{aligned}(-1)^k \binom{-s}{k} &= (-1)^k \frac{(-s)(-s-1)(-s-2)\cdots(-s-k+1)}{k!} \\ &= \frac{s(s+1)(s+2)\cdots(s+k-1)}{k!}\end{aligned}$$

De forma similar, esta formulación del polinomio interpolador se llama polinomio interpolador de Newton con diferencias atrasadas para puntos igualmente espaciados. Se pueden escribir entonces las fórmulas correspondientes a (5.5) para puntos igualmente espaciados. Éstas son

$$f(x) = \sum_{k=0}^{n} \Delta^k y_0 \binom{s}{k} + h^{n+1} \binom{s}{n+1} f^{(n+1)}(\xi), \quad x = x_0 + sh \tag{5.13}$$

y

$$f(x) = \sum_{k=0}^{n} \nabla^k y_n (-1)^k \binom{-s}{k} + h^{n+1} (-1)^{n+1} \binom{-s}{n+1} f^{(n+1)}(\xi), \quad x = x_n + sh \tag{5.14}$$

5.2.4 Interpolación iterativa

El esquema anterior opera en dos etapas; primero se construye la tabla de diferencias divididas para calcular los coeficientes, y después se usa la multiplicación recursiva para encontrar los valores interpolados. Este procedimiento es más conveniente cuando se requiere interpolar valores en un gran número de puntos; pero en el caso en el que se requieren sólo pocos puntos, el gran número de cálculos se puede reducir usando un método de interpolación iterativa, tal como lo es el de Aitken y Neville.

La base de estos métodos es la construcción de una tabla de polinomios de forma similar a la tabla de diferencias divididas, la cual contiene en columnas sucesivas polinomios de más alto orden que ajustan los datos dados en un gran número de puntos cuando uno se mueve a través de la tabla. En el cálculo de los valores interpolados, los polinomios en sí no se usan. Se introduce un valor particular de x y se produce una tabla de valores correspondientes a varios polinomios. El grado del polinomio se incrementa a lo largo de la tabla y, finalmente, se espera que la precisión del resultado se incremente también. De esta forma, si el proceso está convergiendo, el resultado en el lado derecho de la tabla finalmente cumplirá con la precisión requerida. En este punto se detiene el cálculo. La tabla se construye con interpolaciones lineales sucesivas de polinomios anteriores, de acuerdo con la siguiente fórmula:

$$P_{r1,r2,\ldots,rk,i,j}(x) = \frac{(x-x_i)P_{r1,r2,\ldots,rk,j} - (x-x_j)P_{r1,r2,\ldots,rk,i}}{(x_j - x_i)}$$

Inicialmente, $P_0(x) = f_0$, $P_1(x) = f_1$, etc., y los primeros datos de entrada a la tabla de Aitken son

$$P_{01}(x) = \frac{(x-x_1)P_0 - (x-x_0)P_1}{(x_0 - x_1)}$$

$$P_{02}(x) = \frac{(x-x_2)P_0 - (x-x_0)P_2}{(x_0 - x_2)}$$

www.FreeLibros.me

$$P_{012}(x)=\frac{(x-x_2)P_{01}(x)-(x-x_1)P_{02}(x)}{(x_1-x_2)}$$

La tabla 5.3 (de Aitken) completa tiene la siguiente forma:

Tabla 5.3 Tabla de Aitken.

$$\begin{array}{cccccccccc}
x_0 & f_0 & & & & & & & & \\
 & & P_{01}(x) & & & & & & & \\
x_1 & f_1 & & P_{012}(x) & & & & & & \\
 & & P_{02}(x) & & \ddots & & & & & \\
x_2 & f_2 & & & & \ddots & & & & \\
 & & P_{03}(x) & & & & \ddots & & & \\
x_3 & f_3 & & & & & & \ddots & & \\
\dots & \dots & \dots & \dots & \dots\ \dots\ \dots & & & & \ddots & \\
x_k & f_k & P_{0k}(x) & P_{01k}(x) & \dots & \dots & \dots & \dots & \dots & P_{01\cdots k}(x)
\end{array}$$

Si se agrega a esta tabla un valor extra de la función, la nueva columna se calcula usando solamente los valores a lo largo de la diagonal, comenzando por f_0. Por esta razón, sólo los elementos de la diagonal necesitan almacenarse, ya que todos los otros términos son solamente valores intermedios. La tabla de Neville se basa en la misma formulación, pero usa combinaciones de diferentes puntos para formar las columnas. Esto se muestra en la tabla 5.4:

Tabla 5.4 Tabla de Neville.

$$\begin{array}{cccccccccc}
x_0 & f_0 & & & & & & & & \\
 & & P_{01}(x) & & & & & & & \\
x_1 & f_1 & & P_{012}(x) & & & & & & \\
 & & P_{12}(x) & & \ddots & & & & & \\
x_2 & f_2 & & & & \ddots & & & & \\
 & & & & & & \ddots & & & \\
 & & & & & & & \ddots & & \\
\dots & \dots & \dots & \dots & \dots\ \dots\ \dots & & & & \ddots & \\
x_k & f_k & P_{k-1,k}(x) & P_{k-2,k-1,k}(x) & \cdots & \cdots & \cdots & \cdots & \cdots & P_{012\cdots k}(x)
\end{array}$$

Cuando se introduce un nuevo punto en esta tabla, el cálculo involucra sólo elementos del renglón anterior. Por tanto, no se necesita almacenar el resto de la tabla.

El método de interpolación iterativa, o el método de diferencias divididas de Newton, son convenientes para interpolar usando una computadora. En el método iterativo, el grado del término sucesivo de la tabla da fácilmente una guía de la precisión del método; pero cada interpolación necesita el cálculo completo de la tabla de diferencias divididas. Si se van a interpolar muchos puntos, la forma de Newton es más adecuada, ya que la evaluación de la tabla se hace una sola vez. Cada interpolación requiere entonces sólo una multiplicación recursiva, la cual necesita pocos cálculos. Sin embargo, se deben hacer investigaciones preliminares para determinar el número de puntos necesarios para que la interpolación tenga la suficiente precisión.

www.FreeLibros.me

5.3 Elección de los puntos de interpolación

Regresando a la fórmula (5.5),

$$f(x)=P(x)+(x-x_0)(x-x_1)\cdots(x-x_n)\frac{f^{n+1}(\xi)}{(n+1)!} \quad \textbf{(5.14)}$$

Es natural considerar cuál es la mejor elección de los puntos $x_0, x_1,\ldots, x_n$, si esto es posible, para minimizar el error de interpolación,

$$e(x)=(x-x_0)(x-x_1)\cdots(x-x_n)\frac{f^{n+1}(\xi)}{(n+1)!}$$

Una posibilidad es elegir los números $x_0, x_1,\ldots, x_n$ de forma tal que minimicen el máximo del valor absoluto del error de interpolación en el intervalo $[a, b]$. Esto se puede lograr si se minimiza el máximo del valor absoluto de la expresión $(x-x_0)(x-x_1)\cdots(x-x_n)$.

Este problema se conoce como el problema *minimax*. Para efectos prácticos, si se considera que $[a,b]=[-1,+1]$, entonces el problema minimax se puede formular como el de determinar $x_0, x_1,\ldots, x_n$ que minimicen la expresión,

$$\max_{-1\le x\le 1}\left|(x-x_0)(x-x_1)\cdots(x-x_n)\right|$$

Se demostrará que la solución al problema minimax está ligada a los ceros de los polinomios de Tchebyshev, T_{n+1}, los cuales se definen como:

$$T_n(x)=\cos\left(n\cos^{-1}(x)\right) \quad \textbf{(5.15)}$$

Usando la identidad trigonométrica

$$\cos(n+1)\theta+\cos(n-1)\theta=2\cos n\theta\cos\theta$$

se obtiene la relación de recurrencia

$$T_{n+1}(x)=2xT_n(x)-T_{n-1}(x) \quad \textbf{(5.16)}$$

Se tiene de (5.15) que $T_0(x)=1$ y $T_1(x)=x$, y se puede demostrar por inducción que $T_n(x)$ es un polinomio de grado n con coeficiente principal 2^{n-1}. De la relación de recurrencia (5.16) se tiene que

$$T_2(x)=2x^2-1$$

$$T_3(x)=4x^3-3x$$

$$T_4(x)=8x^4-8x^2+1$$

$$T_5(x)=16x^5-20x^3+5x$$

Los ceros de $T_n(x)$ están en el intervalo $[-1, +1]$ y son

$$x_k=\cos\left[\frac{(2k+1)\pi}{2n}\right],\ k=0,1,\ldots,n-1$$

Además, se tiene que $T_n(x)$ alcanza sus máximos y mínimos en los puntos

$$x_k=\cos\left(\frac{k\pi}{n}\right),\ k=1,2,\ldots,n-1$$

Estos valores se encuentran entre los ceros del polinomio. El valor máximo es

$$T_n\left(\cos\left(\frac{k\pi}{n}\right)\right)=(-1)^k$$

www.FreeLibros.me

además de que

$$T_n(-1) = (-1)^n, \; T_n(1) = 1$$

Se tiene también que $|T_n(x)| \leq 1$. Con esta información se procede a resolver el problema minimax mediante el siguiente teorema.

Teorema 5.4 La expresión $\max_{-1 \leq x \leq 1} |(x - x_0)(x - x_1) \cdots (x - x_n)|$ alcanza su valor mínimo cuando los números $x_0, x_1, \ldots, x_n$ se eligen como los ceros del polinomio de Tchebyshev $T_{n+1}(x)$.

Demostración Dado que $T_{n+1}(x)$ tiene coeficiente principal 2^n, se debe demostrar que

$$(x - x_0)(x - x_1) \cdots (x - x_n) = \frac{1}{2^n} T_{n+1}(x)$$

Es decir, de entre todos los polinomios de grado $n+1$ con coeficiente principal igual a 1, el polinomio

$$\frac{1}{2^n} T_{n+1}(x)$$

tiene el módulo máximo más pequeño en el intervalo [−1, +1], el cual es $1/2^n$. Suponiendo que existe un polinomio de $q_{n+1}(x)$ de grado $n+1$ con coeficiente principal igual a 1 tal que

$$\max_{-1 \leq x \leq 1} |q_{n+1}(x)| < \max_{-1 \leq x \leq 1} \frac{1}{2^n} |T_{n+1}(x)| = \frac{1}{2^n},$$

el polinomio $r_n(x) = q_{n+1}(x) - T_{n+1}(x)$ tiene al menos $n+1$ ceros, ya que tiene el mismo signo que $T_{n+1}(x)$ en cada uno de los $n+2$ puntos extremos donde $T_{n+1}(x)$ alcanza su máximo módulo; pero $r_n(x)$ es un polinomio de grado no mayor que n, por lo que no puede tener más de n ceros. Por tanto $r_n(x)$ debe ser idénticamente cero. Esto completa la demostración.

La aparente restricción de su aplicabilidad al intervalo [−1, +1] se puede superar fácilmente por un cambio de variable. Si $X \in [-1, +1]$ y $x \in [a, b]$ entonces la función

$$x = ((X + 1)(b - a)/2) + a$$

transformará el intervalo [−1, +1] al intervalo $[a, b]$. Este cambio de variable es importante en la teoría de cuadratura gaussiana, ya que esto significa que se puede considerar cualquier intervalo finito transformándolo al intervalo [−1, +1]. Como corolario al teorema 5.4, se tiene

Corolario 5.1 Sean $a \leq x_0 < x_1 < x_2 < \cdots < x_n \leq b$ los ceros del polinomio de Tchebyshev T_{n+1}, $f \in C^{(n+1)}[a, b]$ y $f(x_i) = y_i$. Si $P(x)$ es un polinomio interpolador de f en los puntos (x_i, y_i), $i = 0, 1, \ldots, n$, entonces

$$e(x) = |f(x) - P(x)| = \frac{K_{n+1}}{2^n (n+1)!}$$

donde $K_{n+1} = \max_{x \in [a,b]} |f^{(n+1)}(x)|$.

Este corolario establece que, al usar puntos arbitrarios para la interpolación, se tiene convergencia puntual, mientras que al usar puntos dados por las raíces de los polinomios de Tchebyshev se tiene convergencia uniforme. Esto se ilustra con el siguiente ejemplo.

EJEMPLO 5.4

Interpolar la función $f(x) = \exp(-x^2)$ en el intervalo [−5, +5] con 9 puntos de interpolación, usando puntos igualmente espaciados y los ceros del polinomio de Tchebyshev de grado 9.

www.FreeLibros.me

Usando puntos igualmente espaciados se tiene que $n = 8$ y por tanto

$$h = \frac{b-a}{n} = \frac{5-(-5)}{8} = \frac{5}{4}$$

De aquí se obtiene que

$$x_i = a + ih, \;\; x_i = -5, -\frac{15}{4}, -\frac{5}{2}, -\frac{5}{4}, 0, \frac{5}{4}, \frac{5}{2}, \frac{15}{4}, 5$$

Así, el valor de la función en cada punto es

$$f_i = e^{-25}, e^{-\frac{225}{16}}, e^{-\frac{25}{4}}, e^{-\frac{25}{16}}, 1, e^{\frac{25}{16}}, e^{\frac{25}{4}}, e^{\frac{225}{16}}, e^{-25}$$

El polinomio interpolador de Lagrange usando estos puntos es

$$\begin{aligned} P_1(x) = 1 &+ \left(-\frac{1}{875}e^{-25} + \frac{128}{7875}e^{-\frac{225}{16}} - \frac{16}{125}e^{-\frac{25}{4}} + \frac{128}{125}e^{-\frac{25}{16}} - \frac{41}{45}\right)x^2 \\ &+ \left(\frac{28}{28125}e^{-25} - \frac{128}{9375}e^{-\frac{225}{16}} + \frac{2704}{28125}e^{-\frac{25}{4}} - \frac{7808}{28125}e^{-\frac{25}{16}} + \frac{364}{1875}\right)x^4 \\ &+ \left(-\frac{128}{703125}e^{-25} + \frac{512}{234375}e^{-\frac{225}{16}} - \frac{6656}{703125}e^{-\frac{25}{4}} + \frac{14848}{703125}e^{-\frac{25}{16}} - \frac{128}{9375}\right)x^6 \\ &+ \left(\frac{1024}{123046875}e^{-25} - \frac{8192}{123046875}e^{-\frac{225}{16}} + \frac{4096}{17578125}e^{-\frac{25}{4}} - \frac{8192}{17578125}e^{-\frac{25}{16}} + \frac{1024}{3515625}\right)x^8 \end{aligned}$$

Ejecutando las operaciones entre paréntesis se llega finalmente a

$$\begin{aligned} P_1(x) = 1 &- 0.69671613610106987369x^2 + 0.13612707332081008039x^4 \\ &- 0.00924520963764371300 25x^6 + 0.00019403490090295641654x^8 \end{aligned}$$

Por otro lado, los ceros del polinomio de Tchebyshev de grado 9, se determinan con

$$x_i = 5\cos\frac{(2i+1)\pi}{(18)}, \; i = 0, 1, 2, \ldots, n$$

Es decir, se tienen los siguientes 9 ceros,

$$x_i = 5\cos\left(\frac{\pi}{18}\right), \frac{5\sqrt{3}}{2}, 5\cos\left(\frac{5\pi}{18}\right), 5\cos\left(\frac{7\pi}{18}\right), 0, -5\cos\left(\frac{7\pi}{18}\right), -5\cos\left(\frac{5\pi}{18}\right), -\frac{5\sqrt{3}}{2}, -5\cos\left(\frac{\pi}{18}\right)$$

Por tanto la función evaluada en esos puntos es:

$$f_i = e^{-25\cos^2\left(\frac{\pi}{18}\right)}, e^{-\frac{75}{4}}, e^{-25\cos^2\left(\frac{5\pi}{18}\right)}, e^{-25\cos^2\left(\frac{7\pi}{18}\right)}, 1, e^{-25\cos^2\left(\frac{7\pi}{18}\right)}, e^{-25\cos^2\left(\frac{5\pi}{18}\right)}, e^{-\frac{75}{4}}, e^{-25\cos^2\left(\frac{\pi}{18}\right)}$$

donde la transformación $x:[-1,+1] \to [-5,+5]$ definida como $x(t) = 5t$ se usó para transformar las raíces del polinomio de Tchebyshev a los puntos de interpolación. El polinomio interpolador de Lagrange utilizando estos puntos es entonces

$$\begin{aligned} P_2(x) = 1 &- 0.49883156328945086986x^2 + 0.07019796096143466 5691x^4 \\ &- 0.0037043184850804692191x^6 + 0.000065473184005538263827x^8 \end{aligned}$$

De la comparación de $P_1(x)$ y $P_2(x)$ se hace notar que los coeficientes de ambos polinomios son diferentes; la figura 5.4 muestra la comparación de ambos respecto a la función exacta.

www.FreeLibros.me

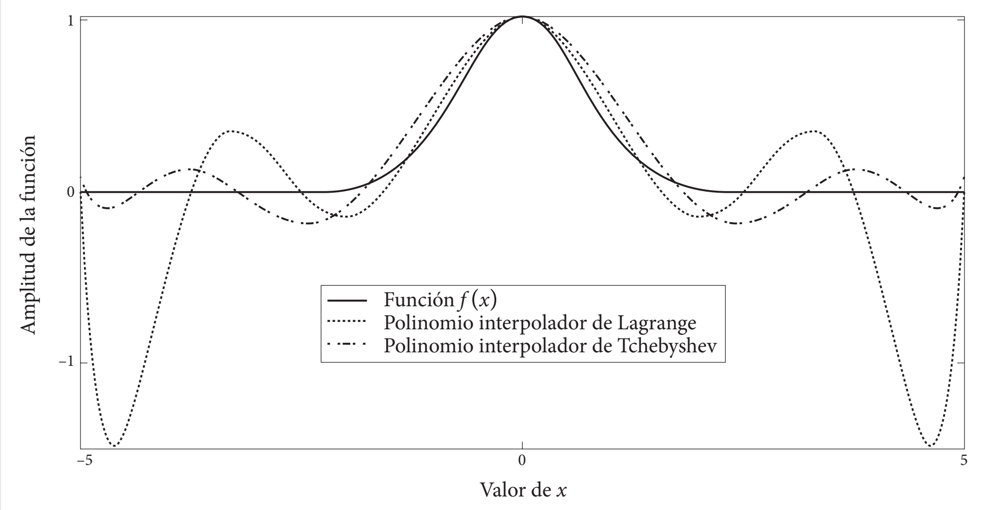


Figura 5.4 Gráfica de la función evaluada y de los dos polinomios interpoladores.

En la figura 5.4, la línea continua representa la función por interpolar; la línea punteada, el polinomio con puntos igualmente espaciados, y la línea punteada doble, el polinomio basado en los ceros del polinomio de Tchebyshev de grado 9. Se puede observar que las oscilaciones dadas por el polinomio, con puntos igualmente espaciados, aumentan al aumentar el número de puntos, ya que la convergencia es puntual, mientras que en el polinomio basado en los puntos de Tchebyshev, estas oscilaciones decrecen al aumentar el número de puntos. En la evaluación numérica de las funciones se utiliza la representación expandida, pues los polinomios de alto orden son muy sensibles a pequeños cambios numéricos de estos coeficientes.

5.4 Ajuste por el método de mínimos cuadrados

La propiedad fundamental de este método es que la suma de los cuadrados de los errores se hace tan pequeña como sea posible [Mathews, 2000], [Burden *et al.*, 2002], [Nieves *et al.*, 2002]. Se tienen dos casos: la aproximación de un conjunto finito de valores o de una función definida dentro de un intervalo. En el primer caso, el error se definirá como la suma de los cuadrados de los errores individuales de $f(x)$ en cada punto; en el segundo caso es necesaria una formulación integral. Esta formulación se usa para demostrar la base teórica del método. En su forma general, el método de mínimos cuadrados se basa en una función aproximada, la cual depende linealmente de un conjunto de parámetros a_0, a_1, … a_n. La integral de la suma del cuadrado de los errores está dada por:

$$S=\int_a^b [f(x)-\phi(a_0,a_1,\ldots,a_n,x)]^2\,dx$$

Debido a que se requiere que S sea lo más pequeño posible, la primera derivada con respecto a los coeficientes debe ser cero esto es:

$$\partial S/\partial a_i=0,\qquad i=0,1,\ldots,n$$

Si las condiciones apropiadas se cumplen para la diferenciación dentro de la integral, entonces ésta se reduce a $n+1$ ecuaciones para los coeficientes a_i. Por tanto se tiene que

$$-2\int_a^b \frac{\partial\phi}{\partial a_i}[f(x)-\phi(a_0,a_1,\ldots,a_n,x)]dx=0,\qquad i=0,1,\ldots,n$$

Como ϕ es función lineal de los coeficientes, los términos de esta ecuación son constantes. Entonces la ecuación se puede escribir como

www.FreeLibros.me

$$\int_a^b \frac{\partial\phi}{\partial a_i}\phi(a_0, a_1, \ldots, a_n, x)dx = \int_a^b \frac{\partial\phi}{\partial a_i} f(x)dx, \qquad i = 0, 1, \ldots, n$$

Esta ecuación se conoce como "la forma normal de la ecuación". Para ilustrar de manera simple el método, considere el caso de una aproximación polinomial,

$$\phi(a_0, a_1, \ldots, a_n, x)dx = a_0 + a_1x + \cdots + a_nx^n$$

La forma normal de esta ecuación es

$$\int_a^b x^i\left[a_0 + a_1x + \cdots + a_nx^n\right]dx = \int_a^b x^i f(x)dx, \qquad i = 0, 1, \ldots, n$$

Si se desarrollan estas ecuaciones, se obtiene

$$\begin{aligned} &u_{00}a_0 + u_{01}a_1 + \cdots + u_{0n}a_n = b_0 \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &u_{n0}a_0 + u_{n1}a_1 + \cdots + u_{nn}a_n = b_n \end{aligned} \tag{5.17}$$

donde los coeficientes u_{ij} del lado izquierdo están dados por

$$u_{ij} = \int_a^b x^i \cdot x^j dx, \qquad i, j = 0, 1, \ldots, n$$

Desafortunadamente, estas ecuaciones pueden ser muy mal condicionadas y no se recomienda la solución directa del problema de mínimos cuadrados por este método. Idealmente, se querrá que la forma normal de la ecuación tome una forma simple que dé una solución eficiente al problema. La forma más simple posible es la forma diagonal, la cual permite poder encontrar los coeficientes a_0, a_1, … a_n directamente, dividiendo por u_{ii} $(i = 0, 1, \ldots, n)$. Esta forma se puede obtener aprovechando las propiedades especiales de las funciones ortogonales. La propiedad fundamental de las funciones ortogonales es que, para dos términos diferentes de una sucesión ortogonal $Q_k(x)$ $(k = 0, 1, \ldots)$, la integral

$$\int_a^b w(x)\cdot Q_i(x)\cdot Q_j(x)dx = 0, \qquad \text{para} \quad i \neq j$$

Los límites a y b y la función de peso $w(x)$ se eligen adecuadamente. En forma específica, si se elige $w(x) = 1$ y $a = -1$, $b = +1$, entonces se obtienen los polinomios de Legendre, los cuales tienen la propiedad de ortogonalidad. Así, en vez de aproximar por sumas de potencias de x, se usa la suma de polinomios de Legendre de la forma

$$\phi(a_0, a_1, \ldots, a_n, x)dx = a_0\overline{P}_0(x) + a_1\overline{P}_1(x) + \cdots + a_n\overline{P}_n(x)$$

Los primeros polinomios de esta serie son

$$\overline{P}_0(x) = 1$$

$$\overline{P}_1(x) = x$$

$$\overline{P}_2(x) = \frac{1}{2}\left(3x^2 - 1\right)$$

Haciendo la correspondencia con la ecuación (5.17), ahora se tiene un conjunto con términos solamente en la diagonal, ya que:

$$u_{ij} = \int_{-1}^{+1} \overline{P}_i(x)\cdot\overline{P}_j(x)dx = 0, \quad i = 0, 1, \ldots, n, \;\; j \neq i$$

Los coeficientes de la diagonal también toman una forma especialmente sencilla,

www.FreeLibros.me

$$u_{ii}=\int_{-1}^{+1}[\overline{P}_i(x)]^2\,dx=\frac{2}{2i+1}$$

de tal forma que los a_i se encuentran directamente:

$$a_i=\frac{2i+1}{2}\int_{-1}^{+1}\overline{P}_i(x)\cdot f(x)dx,\qquad i=0,1,\ldots,n$$

Si se necesita la representación como un polinomio en x, la expresión para $\overline{P}_i(x)$ se puede sustituir dentro de las ecuaciones una vez que se han determinado los valores a_i, y un reagrupamiento dará el polinomio en x.

5.4.1 Ajuste discreto de mínimos cuadrados normalizado

De las propiedades fundamentales de este método expresado en forma discreta, se llega a la siguiente relación:

$$E_r=\sum_{i=1}^{m}(y_i-P_n(x_i))^2 \tag{5.18}$$

donde

$$P_n(x_i)=\sum_{j=0}^{n}a_jx_i^j \tag{5.19}$$

Sustituyendo la ecuación (5.18) en la ecuación (5.19), se obtiene

$$E_r=\sum_{i=1}^{m}\left(y_i-\sum_{j=0}^{n}a_jx_i^j\right)^2$$

Desarrollando esta última expresión, se llega a

$$E_r=\sum_{i=1}^{m}(y_i)^2-2\sum_{i=1}^{m}(y_i)\left(\sum_{j=0}^{n}a_jx_i^j\right)+\sum_{i=1}^{m}\left(\sum_{j=0}^{n}a_jx_i^j\right)^2$$

Reagrupando finalmente, se obtiene

$$E_r=\sum_{i=1}^{m}y_i^2-2\sum_{j=0}^{n}a_j\left(\sum_{i=1}^{m}y_ix_i^j\right)+\sum_{j=0}^{n}\sum_{k=0}^{n}a_ja_k\left(\sum_{i=1}^{m}x_i^{j+k}\right)$$

El error mínimo cuadrado se obtiene cuando la derivada del error respecto a los coeficientes es cero. Por tanto, se debe cumplir que

$$\frac{\partial E_r}{\partial a_j}=0$$

De esa forma se obtiene

$$\frac{\partial\left(\sum_{i=1}^{m}y_i^2-2\sum_{j=0}^{n}a_j\left(\sum_{i=1}^{m}y_ix_i^j\right)+\sum_{j=0}^{n}\sum_{k=0}^{n}a_ja_k\left(\sum_{i=1}^{m}x_i^{j+k}\right)\right)}{\partial a_j}=0$$

Resolviendo la ecuación anterior, se llega a

$$-2\sum_{i=1}^{m}y_ix_i^j+2\sum_{k=0}^{n}a_k\left(\sum_{i=1}^{m}x_i^{j+k}\right)=0,\quad j=0,1,\ldots,n$$

Dividiendo la ecuación previa entre 2 y pasando al lado derecho la sumatoria, se obtiene

$$\sum_{k=0}^{n}a_k\left(\sum_{i=1}^{m}x_i^{j+k}\right)=\sum_{i=1}^{m}y_ix_i^j,\quad j=0,1,\ldots,n$$

www.FreeLibros.me

En forma expandida, la solución es

$$
\begin{array}{ll}
 & \quad k=0 \qquad\quad k=1 \qquad\quad k=2 \qquad\qquad\qquad k=n \\
j=0 & a_0\sum_{i=1}^{m} x_i^0 + a_1\sum_{i=1}^{m} x_i^1 + a_2\sum_{i=1}^{m} x_i^2 + \ldots + a_n\sum_{i=1}^{m} x_i^n = \sum_{i=1}^{m} y_i x_i^0 \\
j=1 & a_0\sum_{i=1}^{m} x_i^1 + a_1\sum_{i=1}^{m} x_i^2 + a_2\sum_{i=1}^{m} x_i^3 + \ldots + a_n\sum_{i=1}^{m} x_i^{n+1} = \sum_{i=1}^{m} y_i x_i^1 \\
j=2 & a_0\sum_{i=1}^{m} x_i^2 + a_1\sum_{i=1}^{m} x_i^3 + a_2\sum_{i=1}^{m} x_i^4 + \ldots + a_n\sum_{i=1}^{m} x_i^{n+2} = \sum_{i=1}^{m} y_i x_i^2 \\
 & \\
j=n & a_0\sum_{i=1}^{m} x_i^n + a_1\sum_{i=1}^{m} x_i^{n+1} + a_2\sum_{i=1}^{m} x_i^{n+2} + \ldots + a_n\sum_{i=1}^{m} x_i^{n+n} = \sum_{i=1}^{m} y_i x_i^n
\end{array}
$$

Reacomodando en forma matricial, se tendrá por tanto

$$
\begin{bmatrix}
\sum_{i=1}^{m} x_i^0 & \sum_{i=1}^{m} x_i^1 & \sum_{i=1}^{m} x_i^2 & \cdots & \sum_{i=1}^{m} x_i^n \\
\sum_{i=1}^{m} x_i^1 & \sum_{i=1}^{m} x_i^2 & \sum_{i=1}^{m} x_i^3 & \cdots & \sum_{i=1}^{m} x_i^{n+1} \\
\sum_{i=1}^{m} x_i^2 & \sum_{i=1}^{m} x_i^3 & \sum_{i=1}^{m} x_i^4 & \cdots & \sum_{i=1}^{m} x_i^{n+2} \\
\vdots & \vdots & \vdots & \ddots & \\
\sum_{i=1}^{m} x_i^n & \sum_{i=1}^{m} x_i^{n+1} & \sum_{i=1}^{m} x_i^{n+2} & \cdots & \sum_{i=1}^{m} x_i^{n+n}
\end{bmatrix}
\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}
=
\begin{bmatrix}
\sum_{i=1}^{m} y_i x_i^0 \\
\sum_{i=1}^{m} y_i x_i^1 \\
\sum_{i=1}^{m} y_i x_i^2 \\
\vdots \\
\sum_{i=1}^{m} y_i x_i^n
\end{bmatrix}
$$

donde m es el número de puntos considerados.

Investigaciones en el campo de la elección del valor adecuado de n dan una visión considerable de la naturaleza de la aproximación de mínimos cuadrados. Cuando $n = m$, se produce un polinomio interpolador para los $n + 1$ puntos usados. Para valores de $n < m$, el ajuste de mínimos cuadrados normalmente no pasará por los puntos, y la curva se sujetará al proceso de suavizamiento. Esto tiene un valor particular cuando el método se aplica a resultados experimentales, los cuales dan valores de la función junto con errores experimentales. La desviación más pequeña debida a los errores puede dar como resultado un polinomio altamente oscilatorio que es esencialmente el reflejo de la fluctuación debida al error. Con curvas que se espera sean suaves, incluso para valores entre dos puntos, la aproximación se puede hacer dividiendo el intervalo en muchos subintervalos. Entonces en cada subintervalo se puede usar una aproximación de mínimos cuadrados basada en un polinomio de orden bajo. La sección 5.9.4 de este capítulo proporciona el código Matlab para hacer un ajuste de hasta (m − 1) orden utilizando la técnica de mínimos cuadrados.

EJEMPLO 5.5

Ajustar la tabla 5.5 de valores por el método de mínimos cuadrados, construir todas las aproximaciones posibles y comparar los resultados.

Tabla 5.5 Tabla de valores para construir los polinomios por mínimos cuadrados.

x	−3	−2	−1	0	1	2	3
$f(x)$	−27	−16	−9	1	8	18	26

www.FreeLibros.me

El polinomio de mayor orden posible por construir con este grupo de datos es de 6, y el menor, por supuesto, de orden cero. Los polinomios finales tienen la siguiente estructura:

$$p(x) = a_0x^0 + a_1x^1 + \cdots + a_nx^n$$

Si se determina primero el de mayor orden, donde $n = 6$, se obtiene el sistema siguiente:

$$\begin{bmatrix} \sum_{i=1}^{m} x_i^0 & \sum_{i=1}^{m} x_i^1 & \cdots & \sum_{i=1}^{m} x_i^6 \\ \sum_{i=1}^{m} x_i^1 & \sum_{i=1}^{m} x_i^2 & & \\ \vdots & & \ddots & \\ \sum_{i=1}^{m} x_i^6 & \sum_{i=1}^{m} x_i^7 & & \sum_{i=1}^{m} x_i^{12} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_6 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{m} y_i x_i^0 \\ \sum_{i=1}^{m} y_i x_i^1 \\ \vdots \\ \sum_{i=1}^{m} y_i x_i^6 \end{bmatrix}$$

Sustituyendo los valores de la tabla en la ecuación anterior se obtiene

$$\begin{bmatrix} 7 & 0 & 28 & 0 & 196 & 0 & 1588 \\ 0 & 28 & 0 & 196 & 0 & 1588 & 0 \\ 28 & 0 & 196 & 0 & 1588 & 0 & 13636 \\ 0 & 196 & 0 & 1588 & 0 & 13636 & 0 \\ 196 & 0 & 1588 & 0 & 13636 & 0 & 120148 \\ 0 & 1588 & 0 & 13636 & 0 & 120148 & 0 \\ 1588 & 0 & 13636 & 0 & 120148 & 0 & 1071076 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} 1 \\ 244 \\ -2 \\ 1720 \\ -50 \\ 13984 \\ -602 \end{bmatrix}$$

El número de coeficientes que se necesitan determinar depende del orden del polinomio que ajusta el grupo de datos; esto se hace particionando la matriz. Por ejemplo, para un polinomio de orden 3, se toma una matriz de cuatro por cuatro para dar

$$\begin{bmatrix} 7 & 0 & 28 & 0 \\ 0 & 28 & 0 & 196 \\ 28 & 0 & 196 & 0 \\ 0 & 196 & 0 & 1588 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 244 \\ -2 \\ 1720 \end{bmatrix}$$

Así, los coeficientes se obtienen simplemente usando la inversa de la matriz,

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 7 & 0 & 28 & 0 \\ 0 & 28 & 0 & 196 \\ 28 & 0 & 196 & 0 \\ 0 & 196 & 0 & 1588 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 244 \\ -2 \\ 1720 \end{bmatrix} = \begin{bmatrix} \frac{3}{7} \\ \frac{1049}{126} \\ -\frac{1}{14} \\ \frac{1}{18} \end{bmatrix}$$

Por tanto, el polinomio de orden 3 para ajustar el grupo de datos es

$$p_3(x) = \tfrac{1}{18}x^3 - \tfrac{1}{14}x^2 + \tfrac{1049}{126}x + \tfrac{3}{7}$$

Todos los polinomios posibles se resumen en la tabla 5.6.

www.FreeLibros.me

Tabla 5.6 Resumen de los polinomios.

Orden	Polinomio
0	$p_0(x)=\frac{1}{7}$
1	$p_1(x)=\frac{61}{7}x+\frac{1}{7}$
2	$p_2(x)=-\frac{1}{14}x^2+\frac{61}{7}x+\frac{3}{7}$
3	$p_3(x)=\frac{1}{18}x^3-\frac{1}{14}x^2+\frac{1049}{126}x+\frac{3}{7}$
4	$p_4(x)=-\frac{1}{22}x^4+\frac{1}{18}x^3+\frac{4}{11}x^2+\frac{1049}{126}x-\frac{3}{77}$
5	$p_5(x)=\frac{1}{120}x^5-\frac{1}{22}x^4-\frac{1}{24}x^3+\frac{4}{11}x^2+\frac{128}{15}x-\frac{3}{77}$
6	$p_6(x)=-\frac{1}{15}x^6+\frac{1}{120}x^5+\frac{5}{6}x^4-\frac{1}{24}x^3-\frac{34}{15}x^2+\frac{128}{15}x+1$

La figura 5.5 muestra la tabla de datos en círculos, y todos sus ajustes de orden par, es decir, de orden 0, 2, 4 y 6. Del análisis de esta figura se puede deducir que no necesariamente al aumentar el orden del polinomio resultante se mejora el ajuste. Esto se explica de manera natural si se tiene un grupo de datos provenientes de una línea recta: en este caso, lo más lógico es que una función lineal de primer orden se aproxime mejor que una de alto orden.

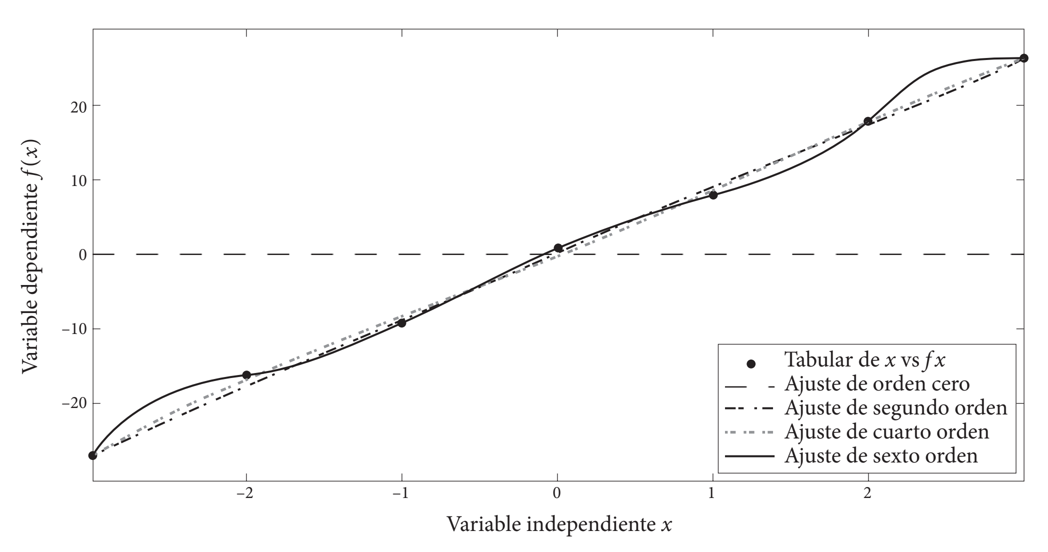


Figura 5.5 Gráfica de la tabla de datos y todas sus funciones de ajuste.

Adicionalmente se tiene el teorema 5.2, el cual enfatiza que con 7 puntos se tiene al máximo un polinomio de sexto orden. Adicionalmente, sin importar la técnica que se utiliza dentro del intervalo, todos los polinomios son equivalentes. Por esta razón, y por las razones expuestas en las secciones 5.1 y 5.2, construir un polinomio interpolador de máximo orden por la técnica de mínimos cuadrados no es lo adecuado. Por supuesto, para este caso específico, se minimizan las propiedades del método.

www.FreeLibros.me

5.5 Transformada rápida de Fourier

Muchos de los obstáculos encontrados en la aplicación del análisis de Fourier a problemas prácticos se superan mediante el uso de transformadas numéricas [Hwei P. Hsu, 1998]. Aunque la transformada de Fourier es una herramienta de análisis muy poderosa, su aplicación a problemas prácticos suele ser muy limitada. Algunas causas de esto son: 1) Funciones del dominio del tiempo $f(t)$ o de la frecuencia $F(s)$ difíciles o imposibles de pasar de un dominio al otro; 2) Funciones de tiempo no especificadas analíticamente, sino por medio de gráficas, de mediciones experimentales o en forma discretizada.

Algunas metodologías se basan en el uso de funciones racionales del dominio de Laplace, o bien en la aproximación a éstas. Aquí, en cambio, se utiliza la transformada discreta de Fourier (TDF). Dicho enfoque es mucho más general, pues permite el manejo de funciones irracionales. Adicionalmente permite determinar, y en buena medida controlar, los niveles máximos de error numérico.

5.5.1 Transformadas de Fourier y de Laplace

Sea $f(t)$ una forma de onda y $F(s)$ su imagen en el dominio de Laplace. Sus transformadas directa e inversa de Laplace son

$$F(s) = \int_0^\infty f(t)\, e^{-st} dt \tag{5.20a}$$

$$f(t) = \frac{1}{2\pi j} \int_{c-j\infty}^{c+j\infty} F(s)\, e^{-st} ds \tag{5.20b}$$

donde $s = c + j\omega$, ω es la frecuencia angular y c una constante finita con valor mayor o igual a cero.

De la sustitución de $s = c + j\omega$ en (5.20a) y (5.20b) se llega a

$$F(c + j\omega) = \int_0^\infty \left[f(t)\, e^{-ct} \right] e^{-j\omega t} dt \tag{5.21a}$$

$$f(t) = \frac{e^{ct}}{2\pi} \int_{-\infty}^{\infty} F(c + j\omega)\, e^{j\omega t} d\omega \tag{5.21b}$$

Cuando c vale cero (5.21a) y (5.21b) corresponden a las transformadas de Fourier:

$$F(j\omega) = \int_0^\infty f(t)\, e^{-j\omega t} dt \tag{5.22a}$$

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(j\omega)\, e^{j\omega t} d\omega \tag{5.22b}$$

El límite inferior en (5.22a), normalmente $-\infty$, aquí se toma como 0, pues sólo se considerarán funciones causales. La expresión (5.21a) indica que la transformada de Laplace se puede obtener aplicando la integral de Fourier a $f(t)$ amortiguada; es decir, premultiplicada por una exponencial decreciente. La constante c es su coeficiente de amortiguamiento. La expresión (5.21b), por su parte, indica que la transformada inversa de Laplace se puede obtener aplicando la integral inversa de Fourier a $F(s)$ y luego desamortiguando el resultado, multiplicándolo por una exponencial creciente de la forma e^{ct}.

5.5.2 Tratamiento numérico de la transformada de Fourier

La integración numérica requiere límites de integración finitos, por lo que el rango de t en (5.22a) se trunca, sustituyéndolo por $[0, T]$. Éste, además, también se discretiza, de modo que

$$t = m\Delta t \rightarrow m = 1, 2, 3, \ldots, M \;\; \text{y} \;\; \Delta t = T/M \tag{5.23}$$

La función $f(t)$ se puede entonces representar mediante una serie de puntos o muestras de la forma $f(m\Delta t)$, como se ilustra en la figura 5.6*a*. Del mismo modo que para t, el rango de integración de ω en (5.22b) se trunca en el intervalo $[-\Omega, +\Omega]$; luego, éste se discretiza de modo que

www.FreeLibros.me

$$\omega = n\Delta\omega \rightarrow n = -N, \ldots, -1, 0, +1, \ldots, N \text{ y } \Delta\omega = \Omega/2N \tag{5.24}$$

La figura 5.6b ilustra el truncamiento y discretización del rango de ω, así como el muestreo de $F(j\omega)$,

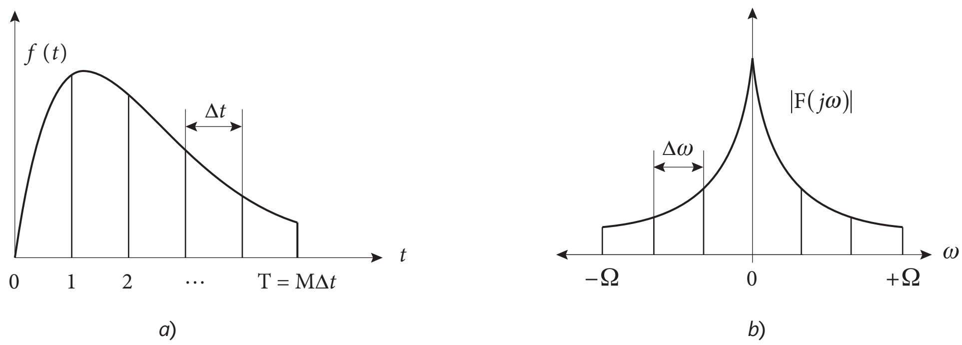


Figura 5.6 Truncamiento, discretización y muestreo. *a*) Para $f(t)$. *b*) Para $F(j\omega)$.

Los rangos muestreados y discretizados de t y de ω permiten evaluar a (5.22a) numéricamente. De la regla rectangular de integración, se obtiene así

$$F(jn\Delta\omega) \cong \Delta t \sum_{m=0}^{M-1} f(m\Delta t)\, e^{-jmn\Delta t\Delta\omega} \tag{5.25a}$$

Se puede demostrar, por sustitución directa, que el lado derecho de (5.25a) es periódico en ω y que su periodo es

$$P_\omega = 2\pi/\Delta t \tag{5.25b}$$

De la aplicación de la regla rectangular de integración en (5.22b), se obtiene

$$f(m\Delta t) \cong \frac{\Delta\omega}{2\pi} \sum_{n=-N}^{N-1} F(jn\Delta\omega)\, e^{jmn\Delta t\Delta\omega} \tag{5.25c}$$

En forma similar a (5.25a), el lado derecho de (5.25c) es periódico en t, y su periodo es

$$P_t = 2\pi/\Delta\omega \tag{5.25d}$$

La repetitividad de (5.25a) implica que 2Ω, la longitud del rango truncado de ω, deba ser menor o igual a P_ω. Se elige la igualdad, pues concuerda con el criterio del muestreo de Nyquist:

$$\Omega = \pi/\Delta t \tag{5.26}$$

El criterio de muestreo de Nyquist establece que

- Si una señal no contiene componentes en frecuencia mayores de $\omega_S \rightarrow \text{rad/s}$, ésta se puede caracterizar por completo con los valores de las muestras tomadas en instantes separados por una frecuencia de muestreo dada por $f_m = \pi/\omega_C \rightarrow \text{s}$, donde ω_C es la frecuencia de la componente de mayor frecuencia contenida en la señal $f(t)$.

La periodicidad de (5.25c) por su parte lleva a la consideración de que T, el tiempo máximo de observación, debe ser menor o igual a P_t. Con el fin de que (5.25c) abarque el mayor intervalo posible de tiempo, también se escoge la igualdad

www.FreeLibros.me

$$T = 2\pi/\Delta\omega \tag{5.27}$$

La combinación algebraica de (5.26) y (5.27) con (5.23) y con (5.24) determina las siguientes relaciones:

$$M = 2N \tag{5.28a}$$

$$\Delta t \Delta\omega = 2\pi/M \tag{5.28b}$$

$$\Delta\omega/2\pi = 1/M\Delta t \tag{5.28c}$$

Como consecuencia adicional de lo anterior, de los parámetros T, Ω, Δt, $\Delta\omega$, y M sólo a dos de ellos (cualesquiera) se les puede asignar valores libremente, pues con esto los demás quedan automáticamente determinados.

Considerando ahora las variables discretas f_m y F_n como aproximaciones correspondientes de $f(m\Delta t)$ y $F(jn\Delta\omega)$, de modo tal que (5.25a) y (5.25c) sean recíprocas, o sea, que la relación de igualdad "=" sustituya a la de aproximación "≅". aplicando además (5.28b) en (5.25a), se llega a

$$F_n = \Delta t \sum_{m=0}^{M-1} f_m e^{-j2\pi mn/M} \tag{5.29a}$$

Para (5.25c) adicionalmente se aplican (5.28a) y (5.28c). Por tanto se tiene que:

$$f_m = \frac{1}{\Delta t}\left[\frac{1}{M}\sum_{n=-M/2}^{M/2-1} F_n e^{j2\pi mn/M}\right] \tag{5.29b}$$

La sumatoria de (5.29a) inmediatamente se identifica con la TDF. En cuanto a (5.29b), es posible demostrar que

$$f_m = \frac{1}{\Delta t}\left[\frac{1}{M}\sum_{n=0}^{M-1} F_n e^{j2\pi mn/M}\right] \tag{5.29c}$$

Ahora (5.29c), se identifica con la TDF inversa. Una importante ventaja de relacionar las transformadas de Fourier con la TDF es la posibilidad de usar el algoritmo de Cooley-Tuckey o FFT. Los errores numéricos de (5.29a) y (5.29c) se deben principalmente a los procesos de truncamiento y de discretización. A continuación se describen los efectos de éstos, así como su control.

5.5.3 Errores por truncamiento

El truncamiento de (5.22b) se puede representar de la siguiente forma:

$$f_1(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} H(j\omega)F(j\omega)\, e^{j\omega t} d\omega \tag{5.30}$$

donde $f_1(t)$ es la aproximación de $f(t)$ por el truncamiento y $H(j\omega)$ es la ventana rectangular.

La ventana rectangular está descrita por

$$H(j\omega) = \begin{cases} 0 & -\Omega > \omega \\ 1 & -\Omega \le \omega \le \Omega \\ 0 & \Omega < \omega \end{cases}$$

La gráfica de $H(j\omega)$ se muestra en la figura 5.7*a*, mientras que en la 5.7*b* se muestra su transformada inversa $h(t)$.

www.FreeLibros.me

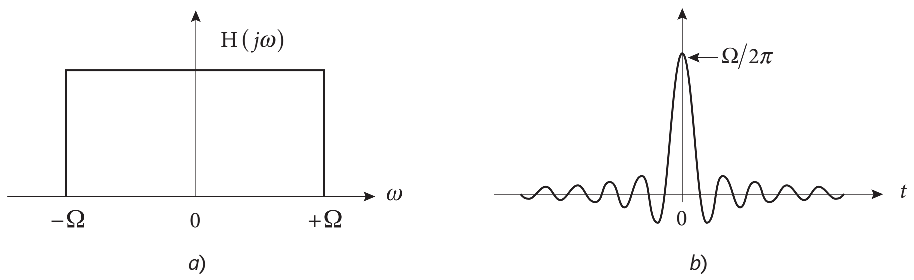


Figura 5.7 Gráficas *a*) de $H(j\omega)$ y *b*) de $h(t)$.

De acuerdo con el teorema de la convolución, $f_1(t)$ en (5.30) es $f(t)\otimes h(t)$. En la figura 5.8*a* se representa $f(t)$ como un escalón. La figura 5.8*b* ilustra el resultado de convolucionar a $f(t)$ con $h(t)$. Ahí se pueden observar dos de los efectos más importantes del truncamiento: 1) Las oscilaciones de Gibbs que se acentúan alrededor del punto de discontinuidad; 2) El adelanto de $\Delta t/2$ segundos de $f_1(t)$ con respecto a $f(t)$.

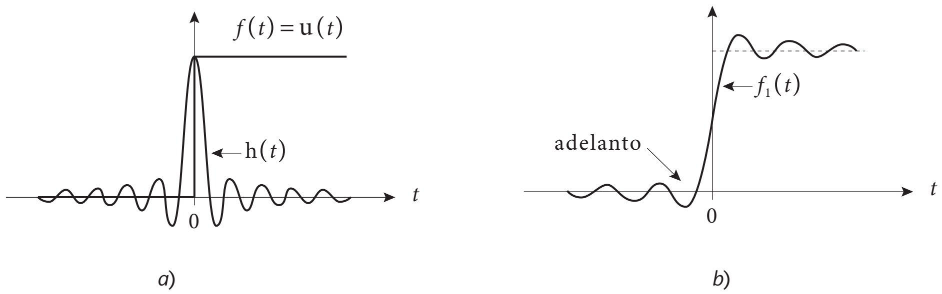


Figura 5.8 Gráficas *a*) de las funciones $f(t)$ y $h(t)$ y *b*) de $f_1(t)$.

Toda función que involucre una discontinuidad se puede expresar como la suma de una función continua y un escalón. De este modo, el ejemplo de la figura 5.8 pasa a ser de interés general. Las oscilaciones de Gibbs se aminoran evitando truncamientos abruptos como el de $H(j\omega)$. Por tanto, una alternativa es cambiar la función $H(j\omega)$ por otra donde el truncamiento va precedido de un suavizamiento. La figura 5.9 muestra dos funciones con esta característica. Funciones como la de las figuras 5.7*a*, 5.9*a* y 5.9*b* se denominan ventanas de datos. La primera es la rectangular; las otras dos son la de Lanczos y la de Von Hann (o Hanning), respectivamente. A fin de evitar errores adicionales debidos al adelanto de $\Delta t/2$ segundos de $f_1(t)$, respecto de $f(t)$ exacta, adicionalmente se aplica a esta ventana un operador de retraso de $\Delta t/2$ segundos.

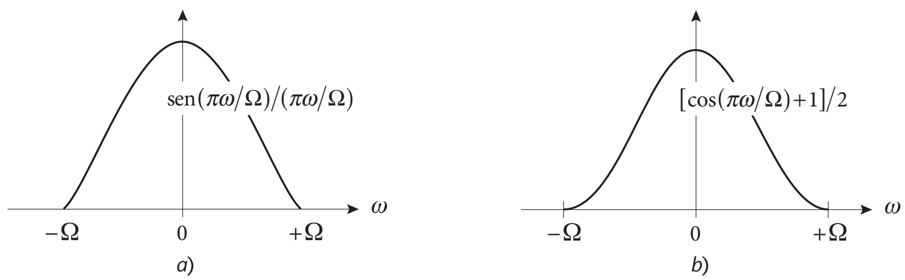


Figura 5.9 Ventana de datos *a*) Lanczos y *b*) Von Hann.

www.FreeLibros.me

5.5.4 Errores por discretización

Considerando ahora la discretización de (5.22b) sin truncamiento,

$$f_2(t)=\frac{\Delta\omega}{2\pi}\sum_{n=-\infty}^{+\infty}F(jn\Delta\omega)e^{jn\Delta\omega t} \tag{5.31}$$

donde $f_2(t)$ es la aproximación resultante. Del teorema del muestreo se obtiene que:

$$F(jn\Delta\omega)e^{jn\Delta\omega t}=\int_{-\infty}^{+\infty}F(j\omega)e^{j\omega t}\delta(\omega-n\Delta\omega)d\omega \tag{5.32}$$

donde $\delta(\omega-n\Delta\omega)$ es la función impulso de Dirac.

Sustituyendo (5.32) en (5.31) e intercambiando el orden de la sumatoria y de la integral se llega a

$$f_2(t)=\frac{1}{2\pi}\int_{-\infty}^{+\infty}F(j\omega)e^{j\omega t}\left\{\Delta\omega\sum_{n=-\infty}^{+\infty}\delta(\omega-n\Delta\omega)\right\}d\omega \tag{5.33}$$

La expresión entre llaves en (5.33) es la transformada de Fourier de un tren periódico de impulsos de Dirac:

$$\delta_T(t)=\sum_{n=-\infty}^{+\infty}\delta(t-nT) \tag{5.34}$$

cuyo periodo es $T=2\pi/\Delta\omega$. Del teorema de la convolución entonces se tiene que

$$f_2(t)=\sum_{n=-\infty}^{+\infty}f(t-nT) \tag{5.35}$$

Esta expresión indica que dentro del rango $0\le t\le T$, $f_2(t)$ está constituida por encimamientos de $f(t)$ y sus desplazamientos $f(t+T)$, $f(t+2T)$, etc., como se muestra en la figura 5.10.

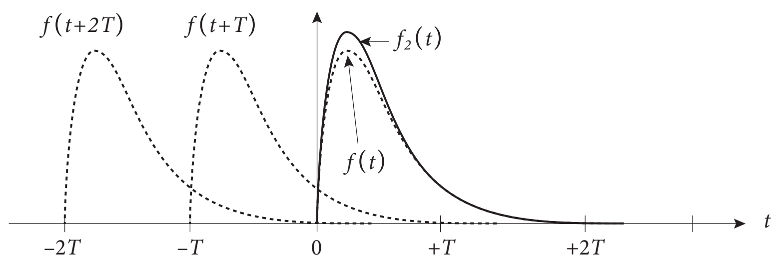


Figura 5.10 Representación $f_2(t)$ como un encimamiento de $f(t)$ y sus desplazamientos.

Es claro en la figura 5.10 que el error de discretización será pequeño en la medida en que $f(t)$ tienda a desvanecerse para valores de t mayores a T. Si $f(t)$ no tiene esta propiedad, se puede amortiguar artificialmente multiplicándola por la exponencial decreciente e^{-ct}.

Este proceso, de hecho, convierte a (5.29a) y a (5.29c) en las transformadas numéricas de Laplace:

$$F_n=\Delta t\sum_{m=0}^{M-1}f'_m e^{-j2\pi mn/M} \tag{5.36a}$$

$$f_m=\frac{e^{cm\Delta t}}{\Delta t}\left[\frac{1}{M}\sum_{n=0}^{M-1}F'_n e^{j2\pi mn/M}\right] \tag{5.36b}$$

En (5.36a) f'_m denota a $f_m e^{-cm\Delta t}$, y en (5.36b) F'_n denota a F_nH_n, siendo H_n una ventana de datos discretizada (Lanczos, Hamming, Von Hann, u otra).

www.FreeLibros.me

Para minimizar el error de encimamiento de $f_2(t)$ en (5.31), convendría dar al coeficiente c el máximo valor posible. Pero, por otro lado, se debe considerar que en (5.36b) este coeficiente juega el papel de amplificador y, dado que las ventanas de datos no eliminan totalmente las oscilaciones de Gibbs, éstas se amplificarían tras la multiplicación por $e^{cm\Delta t}$.

Se define al error entre $f(m\Delta t)$ y f_m de la siguiente forma:

$$\varepsilon = \frac{f(m\Delta t) - f_m}{f_{\text{máx}}(t)}$$

siendo $f_{\text{máx}}(t)$ el valor máximo de $f(t)$. Cuando ε es menor a 0.01, se puede utilizar la siguiente expresión para determinar el correspondiente valor de c:

$$c = -\frac{\log_e(\varepsilon)}{T}$$

5.5.5 Transformada discreta de Fourier como método de ajuste

Si se utiliza la técnica de la transformada de Fourier en forma discreta para ajustar un grupo de datos, se tendrá un grupo de funciones finito que forman una base ortogonal discreta; es decir, si se sustituye la variable independiente (en forma discreta) en las funciones, éstas reproducen la forma de onda discreta. Esto quiere decir que no se puede crear información en forma adicional, y que los datos que se tienen son los datos que se mantienen. Esto explica por qué, para una función continua, se obtiene una base ortogonal de Fourier continua, es decir, un grupo de funciones que ajustan todo el intervalo. El ejemplo 5.6 muestra con claridad cómo se realiza el ajuste en forma discreta, o mejor dicho cómo se utiliza la Transformada Discreta de Fourier para realizar un ajuste discreto mediante la base ortogonal seno-coseno. La sección 5.9.5 de este capítulo provee el código Matlab para realizar el ajuste discreto con la técnica de Fourier para cualquier función discreta.

EJEMPLO 5.6

Si se tiene la función discreta mostrada en la tabla 5.7, hacer la descomposición en la base ortogonal de Fourier, utilizando la TDF.

Tabla 5.7 Tabla discreta de valores que definen la función a aproximar.

Función discreta		Descomposición de Fourier con $n = 0, 1, \ldots, 15$	
t	$f(t)$	Funciones coseno	Funciones seno
0	1.0000	18.5424 cos(0)	18.5424 sen(0)
0.1	0.9058	1.8934 cos($2\pi n\Delta\varphi_1$)	4.0586 sen($2\pi n\Delta\varphi_1$)
0.2	0.8267	–0.2235 cos($4\pi n\Delta\varphi_2$)	1.6207 sen($4\pi n\Delta\varphi_2$)
0.3	0.7677	–0.4405 cos($6\pi n\Delta\varphi_3$)	0.9735 sen($6\pi n\Delta\varphi_3$)
0.4	0.7336	–0.5161 cos($8\pi n\Delta\varphi_4$)	0.6465 sen($8\pi n\Delta\varphi_4$)
0.5	0.7289	–0.5506 cos($10\pi n\Delta\varphi_5$)	0.4312 sen($10\pi n\Delta\varphi_5$)
0.6	0.7571	–0.5680 cos($12\pi n\Delta\varphi_6$)	0.2671 sen($12\pi n\Delta\varphi_6$)
0.7	0.8202	–0.5764 cos($14\pi n\Delta\varphi_7$)	0.1282 sen($14\pi n\Delta\varphi_7$)
0.8	0.9185	–0.5789 cos($16\pi n\Delta\varphi_8$)	0.0000 sen($16\pi n\Delta\varphi_8$)
0.9	1.0499	–0.5764 cos($18\pi n\Delta\varphi_9$)	–0.1282 sen($18\pi n\Delta\varphi_9$)
1.0	1.2094	–0.5680 cos($20\pi n\Delta\varphi_{10}$)	–0.2671 sen($20\pi n\Delta\varphi_{10}$)
1.1	1.3895	–0.5506 cos($22\pi n\Delta\varphi_{11}$)	–0.4312 sen($22\pi n\Delta\varphi_{11}$)
1.2	1.5807	–0.5161 cos ($24\pi n\Delta\varphi_{12}$)	–0.6465 sen($24\pi n\Delta\varphi_{12}$)

www.FreeLibros.me

Función discreta			Descomposición de Fourier con $n = 0, 1, \ldots, 15$	
t	$f(t)$		Funciones coseno	Funciones seno
1.3	1.7724		$-0.4405 \cos(26\,\pi\, n\Delta\varphi_{13})$	$-0.9735\ \text{sen}(26\,\pi\, n\Delta\varphi_{13})$
1.4	1.9558		$-0.2235 \cos(28\,\pi\, n\Delta\varphi_{14})$	$-1.6207\ \text{sen}(28\,\pi\, n\Delta\varphi_{14})$
1.5	2.1262		$1.8934 \cos(30\,\pi\, n\Delta\varphi_{15})$	$-4.0586\ \text{sen}(30\,\pi\, n\Delta\varphi_{15})$

Para la tabla 5.7, los incrementos de ángulo se calculan de la siguiente manera:

$$\Delta\varphi_1 = \frac{2\pi}{Ns}, \quad Ns \text{ es el número de muestras}$$

$$\Delta\varphi_2 = 2\Delta\varphi_1$$

$$\Delta\varphi_3 = 3\Delta\varphi_1$$

$$\ldots$$

$$\Delta\varphi_{Ns-1} = (Ns-1)\Delta\varphi_1$$

Por ejemplo, la muestra número 15 se calcula de la siguiente manera:

- Se calculan todos los $\Delta\varphi_i$.
- Se toma el valor de $n = 14$.
- Se sustituyen los valores en todas las funciones coseno y seno.
- Todos los valores obtenidos se suman para dar $f = 1.95580848472978$. Comparando con la función original tabulada $f(t) = 1.9558$, se tiene un error del orden de $\varepsilon = 1e^{-5}$.

La figura 5.11 muestra la función tabulada y el resultado de calcular los puntos mediante los resultados obtenidos utilizando la Transformada Discreta de Fourier (TDF). De esta forma se obtiene una forma analítica de representar una función que está en forma discreta; por supuesto, se debe hacer notar que el ajuste es válido sólo en los puntos de discretización y que, fuera de ellos, no se tiene certeza de los valores que puedan surgir si se toma la expansión discreta de Fourier, es decir, si se utilizan las funciones para calcular datos fuera de los puntos de discretización.

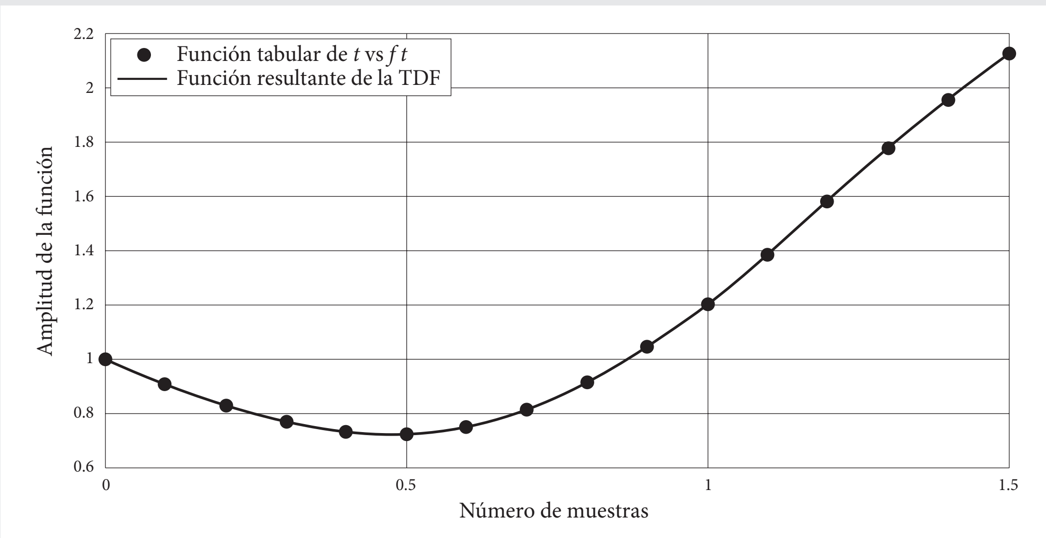


Figura 5.11 Descomposición discreta utilizando la TDF.

www.FreeLibros.me

5.6 Polinomios ortogonales

En vista de la importancia de los polinomios ortogonales en problemas de aproximación, se resumen aquí algunas de las propiedades más importantes de estos polinomios. Se tiene una sucesión de polinomios $Q_k(x)$ de grado k para $k \leq K$, para algún valor fijo de K.

5.6.1 Relación de ortogonalidad

Los polinomios $Q_k(x)$ se dicen ortogonales con respecto a la función de peso $w(x)$ en un intervalo $[a, b]$ si

$$\int_a^b w(x) \cdot Q_i(x) \cdot Q_j(x) dx = 0, \qquad i \neq j$$

donde $w(x) \geq 0$ y $a \leq x \leq b$. Si adicionalmente se satisface la condición siguiente, los polinomios forman un conjunto ortonormal:

$$\int_a^b w(x) \cdot Q_i^2(x) dx = 1 \qquad \text{para todo } i$$

Se debe notar que esta definición implica que un polinomio $Q_i(x)$ es ortogonal a cualquier polinomio de grado menor a i. Esto es porque cualquier polinomio se puede expresar como una combinación lineal de elementos del conjunto $Q_k(x)$ $(k = 0, 1, \ldots, i)$ debido a que son linealmente independientes. Así se tiene que

$$P_j(x) = \sum_{k=0}^{j} a_k \cdot Q_k(x), \qquad j < i$$

y por tanto se tiene que

$$\int_a^b w(x) \cdot Q_i(x) \cdot P_j(x) dx = \sum_{k=0}^{j} a_k \int_a^b w(x) \cdot Q_i(x) \cdot Q_k(x) dx = 0$$

Todos los términos son cero por la propiedad de ortogonalidad. Por tanto, $P_j(x)$ es ortogonal a $Q_i(x)$, $j < i$. Esta propiedad de ortogonalidad da la forma de generar polinomios sucesivos de la serie, aunque la propiedad de recurrencia da un método más adecuado para la implementación práctica.

EJEMPLO 5.7

Como un ejemplo, los coeficientes de un polinomio de Legendre de segundo orden se encuentran de la siguiente forma. La relación de ortogonalidad para el polinomio $ax^2 + bx + c$ da

$$\int_{-1}^{+1} (ax^2 + bx + c) \cdot 1 \cdot dx = 0, \qquad \frac{2a}{3} + 2c = 0$$

y

$$\int_{-1}^{+1} (ax^2 + bx + c) \cdot x \cdot dx = 0, \qquad \frac{2b}{3} = 0$$

De aquí, el polinomio de Legendre de grado 2 es

$$c(-3x^2 + 1)$$

La propiedad de ortogonalidad se cumple para todos los valores de c, y la elección de esta constante depende de un conocimiento profundo de la teoría de las funciones ortogonales. La forma común de este polinomio de Legendre es

$$(1/2)(3x^2 - 1)$$

www.FreeLibros.me

pero si se requiere un conjunto ortonormalizado, el polinomio toma la forma

$$\left(\sqrt{5}/2\sqrt{2}\right)\left(3x^2-1\right)$$

5.6.2 Relación de recurrencia

Se puede ver que el método anterior para encontrar los coeficientes se vuelve impráctico para polinomios de alto orden, ya que involucra la solución de un conjunto muy grande de ecuaciones simultáneas. Afortunadamente, otra propiedad de las funciones ortogonales da un método de generación sencillo y adecuado para el uso por computadora. Todos los conjuntos de polinomios ortogonales satisfacen la relación de recurrencia de la siguiente forma:

$$Q_{n+1}(x)=(x-A_n)Q_n(x)+C_nQ_{n-1}(x), \qquad n=1,2,\ldots$$

donde $C_0=0$ y,

$$A_n=\frac{\int_a^b w(x)x\left(Q_n(x)\right)^2 dx}{\int_a^b w(x)\left(Q_n(x)\right)^2 dx}, \qquad n=1,2,\ldots$$

y

$$C_n=\frac{\int_a^b w(x)\left(Q_n(x)\right)^2 dx}{\int_a^b w(x)\left(Q_{n-1}(x)\right)^2 dx}, \qquad n=1,2,\ldots$$

Así, si se conocen los primeros dos polinomios del conjunto, el tercero se puede encontrar con la ecuación anterior, y entonces el segundo y el tercero se usan para generar el cuarto miembro y así sucesivamente. Esta construcción genera polinomios con coeficiente principal igual a 1. Para los polinomios de Legendre, la relación de recurrencia es:

$$\overline{P}_{n+1}(x)=\frac{2n+1}{n+1}(x)\overline{P}_n(x)-\frac{n}{n+1}\overline{P}_{n-1}(x)$$

Los primeros dos miembros de la sucesión son $\overline{P}_0(x)=1$, $\overline{P}_1(x)=x$. Por tanto,

$$\overline{P}_2(x)=\frac{3}{2}(x)(x)-\frac{1}{2}(1)=\frac{1}{2}\left(3x^2-1\right)$$

$$\overline{P}_3(x)=\frac{5}{3}(x)\left(\frac{1}{2}\left(3x^2-1\right)\right)-\frac{2}{3}(x)=\frac{1}{2}\left(5x^3-3x\right)$$

5.6.3 Ortogonalidad discreta

En los párrafos anteriores se restringió la atención al caso donde se está aproximando una función continua y por consiguiente la propiedad de la integral de ortogonalidad es adecuada. Sin embargo, si se tienen sólo un conjunto de valores discretos de la función, entonces es más apropiado definir el error sólo en términos de estos puntos como

$$S_m=\sum_{i=0}^{m}\left[\phi_n(x_i)-f_i\right]^2$$

Se debe tener en cuenta que, si $Q_n(x)$ es un polinomio de grado n, entonces tiene $n+1$ coeficientes como parámetros variables, y entonces existe una solución única al problema de mínimos cuadrados para $m\geq n+1$. La aproximación típica de mínimos cuadrados usualmente tiene más puntos m que el número de coeficientes del polinomio $n+1$. Si nuevamente se usa una expansión en términos de los polinomios ortogonales, se tiene

www.FreeLibros.me

$$\phi_n(x) = \sum_{k=0}^{n} a_k Q_k(x)$$

Usando la condición de minimización

$$\frac{\partial S_m}{\partial a_r} = 0,$$

se genera la ecuación normal

$$\sum_{k=0}^{n} a_k \sum_{i=0}^{m} Q_k(x_i) Q_r(x_i) = \sum_{i=0}^{m} Q_r(x_i) f_i, \qquad r = 0, 1, 2, \ldots, n$$

Si la función satisface la propiedad de ortogonalidad discreta

$$\sum_{i=0}^{m} Q_k(x_i) Q_r(x_i) = 0, \qquad k \neq r,$$

entonces se produce un conjunto diagonal de ecuaciones que tiene como solución

$$a_r = \frac{\sum_{i=0}^{m} Q_r(x_i) f_i}{\sum_{i=0}^{m} Q_r^2(x_i)}, \quad r = 0, 1, 2, \ldots, n$$

Todos los polinomios ortogonales estándar tienen un conjunto de puntos x_i $(i = 0, \ldots, m)$ sobre los cuales se mantiene la propiedad de ortogonalidad discreta. Los polinomios que se usan para ilustrar estas propiedades son los polinomios de Tchebyshev, los cuales son también de interés en aproximación minimax. Para un valor fijo de N, todos los polinomios de Tchebyshev $T_n(x)(n < N)$ tienen la propiedad de ortogonalidad discreta con respecto al conjunto de puntos:

$$x_j = \cos\left[\frac{\pi j}{n}\right], \qquad j = 0, 1, \ldots, N \tag{5.37}$$

El problema principal asociado con el uso de la aproximación discreta de mínimos cuadrados es que se debe ser capaz de obtener los valores de la función en cualquier punto x determinado por la fórmula (5.37), y esto no siempre puede ser posible. También, hay dificultades si es necesario incrementar el orden de la función aproximada. En el caso continuo, el cálculo simplemente involucra encontrar las cantidades

$$a_k = \frac{\int_a^b Q_k(x) f(x) dx}{\int_a^b Q_k^2(x) dx}$$

y, para calcular más valores de k, no se afectan los valores previos.

En el caso discreto, la sumatoria involucra el cálculo de f_i en un nuevo conjunto de valores de x, los cuales en general son diferentes para valores diferentes de m. Una de las ventajas de la formulación de Tchebyshev es que los x_j de orden m también están en la formulación de $2m$, $4m$, etc., entonces, algunos de los f_i son conocidos a partir de valores calculados previamente. Otro punto en el cual se necesita pensar es la elección de los valores n y m. En general, si el número de puntos m se incrementa, esto dará una aproximación más cercana a expensas de mayores recursos de cómputo; se debe elegir una solución intermedia entre precisión y tiempo de cómputo. La elección de un valor adecuado depende de cada problema en particular y no se puede dar una regla que sea general.

5.6.4 Raíces de los polinomios

Si se tiene un conjunto de polinomios

$$Q_k(x) \ (k = 0, 1, \ldots)$$

www.FreeLibros.me

ortogonal en el intervalo $[a, b]$, entonces las raíces

$$x_j \ (j = 0, 1, \ldots, k)$$

de $Q_k(x) = 0$ son todas reales y distintas y están en el intervalo $a < x < b$.

Esta propiedad es necesaria cuando se usan las raíces de un polinomio ortogonal como los puntos base para una aproximación discreta de mínimos cuadrados. Sin embargo, si hubiera raíces repetidas o complejas, no sería posible usarlas en el método de mínimos cuadrados.

5.6.5 Polinomios ortogonales importantes

Se puede ver claramente cómo los polinomios de Legendre surgen en forma natural en la teoría de mínimos cuadrados si las propiedades de ortogonalidad involucran funciones de peso unitarias. La aparente restricción de su aplicabilidad al intervalo $[-1, +1]$ se puede fácilmente superar por un cambio de variable. Si $X \in [-1, +1]$ y $x \in [a, b]$, entonces la ecuación $x = ((X+1)(b-a)/2) + a$ efectuará la transformación al intervalo $[-1, +1]$. Este cambio de variable es importante en la teoría de cuadratura gaussiana, ya que esto significa que se puede considerar cualquier intervalo finito. Sin embargo, las transformaciones como la anterior no pueden incluir el intervalo semiinfinito $[0, \infty]$ o el intervalo infinito $[-\infty, +\infty]$. Los polinomios de Laguerre son ortogonales en el intervalo $[0, \infty]$ y se pueden usar para la integración gaussiana en este intervalo. Los polinomios de Hermite son ortogonales en el intervalo $[-\infty, +\infty]$.

Otro conjunto de polinomios ortogonales son los polinomios de Tchebyshev, los cuales son ortogonales en el intervalo $[-1, +1]$ con la función de peso $(1 - x^2)^{-1/2}$. La propiedad más significativa de estos polinomios es su propiedad de oscilaciones iguales; es decir, todos los máximos y mínimos de la función ocurren en el intervalo $[-1, +1]$ y tienen la misma magnitud absoluta. Esto conduce a la propiedad minimax. Algunas de la propiedades de los polinomios mencionados se resumen en la tabla 5.8.

Tabla 5.8 Funciones de peso de los polinomios ortogonales más conocidos, intervalo en el cual son válidos, sus tres primeros términos y su relación de recurrencia. En la tabla, la correspondencia es 1) Jacobi, 2) Legendre, 3) Tchebyshev, 4) Laguerre y 5) Hermite.

	$w(x)$	a	b	1o.	2o.	3o.	Relación de recurrencia
1	$(1-x)^{-p}(1+x)^{-q}$ $p<1 \quad q<1$	-1	$+1$				
2	$1, p=q=0$	-1	$+1$	1	x	$(3x^2-1)/2$	$P_{n+1}(x) = \frac{2n+1}{n+1} x P_n(x) - \frac{n}{n+1} P_{n-1}(x)$
3	$(1-x^2)^{-1/2}, p=q=\frac{1}{2}$	-1	$+1$	1	x	$2x^2-1$	$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$
4	$e^{-\alpha x}$	0	∞	1	$1-x$	$2-4x+x^2$	$L_{r+1}(x) = (1+2r-x)L_r(x) - r^2 L_{n-1}(x)$
5	$e^{-\alpha^2 x^2}$	$-\infty$	$+\infty$	1	$2x$	$4x^2-2$	$H_{r+1}(x) = 2xH_r(x) - 2rH_{n-1}(x)$

5.7 Polinomios de Tchebyshev y aproximación minimax

Los polinomios de Tchebyshev son un caso especial de los polinomios de Jacobi con $p = q = -1/2$. Son ortogonales en el intervalo $[-1, +1]$ con la función de peso $(1-x^2)^{-1/2}$. El grupo de polinomios conocidos como polinomios de Tchebyshev se definen como [Maron, 1995], [Mathews, 2000], [Burden *et al.*, 2002]:

$$T_n(x) = \cos(n \cos^{-1} x), \qquad -1 \leq x \leq +1$$

www.FreeLibros.me

Esta liga con las funciones trigonométricas es una ayuda útil para probar algunas de las propiedades de los polinomios de Tchebyshev; la reformulación del problema en términos de θ a menudo conduce a una prueba simple. Por ejemplo, la propiedad de ortogonalidad establece que:

$$\int_{-1}^{+1} \frac{T_r(x)T_s(x)dx}{\left(1+x^2\right)^{1/2}} = 0, \qquad r \neq s$$

Haciendo la transformación $x = \cos(\theta)$, se tiene $dx = -\text{sen}\theta d\theta$ y la integral se transforma en

$$\int_{-\pi}^{0} \cos r\theta \cos s\theta d\theta = \frac{1}{2}\int_{-\pi}^{0} \cos[(r+s)\theta] + \cos[(r-s)\theta]d\theta$$

La integral de cos $n\theta$ donde n es un entero dentro del intervalo $(-\pi \leq \theta \leq 0)$ es cero a menos que $n=0$, para que ambas partes de la integral se vuelvan cero, excepto cuando $r = s$. Así, la condición de ortogonalidad está probada. La relación de recurrencia para los polinomios de Tchebyshev se puede derivar por la misma metodología, si

$$T_{n+1}(x) \equiv \cos(n+1)\theta = \cos n\theta \cos\theta - \text{sen}\ n\theta\ \text{sen}\theta$$

$$T_{n-1}(x) \equiv \cos(n-1)\theta = \cos n\theta \cos\theta + \text{sen} n\theta\ \text{sen}\theta$$

Así,

$$T_{n+1}(x) + T_{n-1}(x) = 2x\cos n\theta$$

o bien

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \tag{5.38}$$

donde

$$x = \cos\ \theta$$

$$T_k(x) = \cos\ k\theta$$

5.7.1 La propiedad minimax

En la sección 5.3 se demostró que los polinomios de Tchebyshev conducen a la solución del problema minimax (véase figura 5.4). La elección de los puntos de interpolación es importante, ya que si se requieren $n+1$ puntos, es necesario que estos puntos sean las raíces del polinomio T_{n+1}. En el caso de que se considere el intervalo $[a, b]$, es necesario transformar dicho intervalo al intervalo $[-1, +1]$ por medio de la transformación ya señalada.

5.7.2 Economización de polinomios

Las anteriores propiedades de los polinomios de Tchebyshev se pueden usar para encontrar la mejor aproximación polinomial de grado $n - 1$ para un polinomio dado de grado n. Si el polinomio dado es

$$p_n(x) = a_0 + a_1 x + \ldots + a_n x^n, \qquad -1 \leq x \leq +1,$$

entonces se forma el polinomio

$$q_{n-1}(x) = p_n(x) - \frac{a_n}{2^{n-1}} T_n(x)$$

Se puede notar que $q_{n-1}(x)$ es un polinomio con un grado máximo $n-1$, ya que el coeficiente de x^n es cero. La diferencia entre los polinomios $q_{n-1}(x)$ y $p_n(x)$ es un múltiplo de un polinomio de Tchebyshev, y

www.FreeLibros.me

por consiguiente se desvía de cero en lo mínimo dentro del intervalo $[-1,+1]$. El proceso de economización se refiere a quitar el término de mayor grado restando un múltiplo del polinomio de Tchebyshev apropiado. Cuando se ha encontrado un nuevo polinomio $q_{n-1}(x)$, es posible entonces formar la mejor aproximación a éste por el mismo método.

$$q_{n-2}(x)=q_{n-1}(x)-\frac{b_{n-1}}{2^{n-2}}T_{n-1}(x)$$

Cada término de la sucesión será la mejor aproximación al término anterior; pero se debe comprender que la aproximación de $p_n(x)$ será sólo una buena aproximación, en el sentido minimax, por el polinomio $q_{n-1}(x)$. Por ejemplo, el error de la segunda aproximación está dado por

$$q_{n-2}(x)-p_n(x)=-\frac{b_{n-1}}{2^{n-2}}T_{n-1}(x)-\frac{a_n}{2^{n-1}}T_n(x)$$

y la suma de dos polinomios de Tchebyshev no posee la propiedad minimax. No obstante, se ha encontrado en la práctica que este proceso de economización da una buena aproximación de bajo orden. El error máximo posible de la aproximación se encuentra fácilmente debido a que $T_r(x)$ tiene un máximo y un mínimo de ±1. En consecuencia, para $q_{n-2}(x)$, se tiene un error con un módulo máximo menor que:

$$\left|\frac{b_{n-1}}{2^{n-2}}\right|+\left|\frac{a_n}{2^{n-1}}\right|$$

5.7.3 Expansión en series de Tchebyshev

Como una alternativa a la expansión en series de Taylor seguida de economización [Nakamura,1992], [Mathews, 2000], es posible extender directamente una expansión en series de Tchebyshev. Cualquier función continua $f(x)$ con un número finito de máximos y mínimos en el intervalo $[-1,+1]$ se puede expandir de la siguiente manera,

$$f(x)=\sum_{k=0}^{\infty}{}' a_kT_k(x) \tag{5.39}$$

El apóstrofo significa que el primer término es $a_0/2$, lo cual simplifica la definición de los coeficientes a_r de la misma manera que los coeficientes de Fourier. Con la propiedad de ortogonalidad, los coeficientes se pueden evaluar multiplicando la ecuación (5.39) por

$$\left[\left(1+x^2\right)^{-1/2}\cdot T_s(x)\right]$$

e integrando entre los límites $[-1,+1]$. Todos los términos del lado derecho, excepto uno, serán cero. Esto conduce a

$$a_s=\frac{2}{\pi}\int_{-1}^{+1}\frac{T_s(x)f(x)dx}{\left(1+x^2\right)^{1/2}}$$

Sería, por supuesto, necesario evaluar esta integral por medios aproximados si se usan medios computacionales. Es preferible tener un esquema basado en series finitas, usando las propiedades de ortogonalidad discretas. Si se supone que los valores de $f(x)$ son conocidos en los puntos

$$x_j=\cos[\pi j/n](j=0,1,\ldots,n)$$

y se elige la función

$$\phi(x)=\sum_{k=0}^{n}{}'' b_kT_k(x)$$

para que los valores coincidan en los puntos $x_j(j=0,1,\ldots,n)$, entonces

www.FreeLibros.me

$$f(x_j) = \phi(x_j) = \sum_{k=0}^{n}{}'' b_k T_k(x_j), \qquad j = 0, 1, \ldots, n \tag{5.40}$$

El $\sum''$ indican que el primero y último término tienen coeficientes $b_0/2$ y $b_n/2$ respectivamente.

5.7.4 Ortogonalidad discreta

La propiedad de ortogonalidad discreta está definida por

$$\sum_{j=0}^{n}{}'' T_r(x_j) T_s(x_j) = \begin{cases} n & r = s = 0, n \\ n/2 & r = s \neq 0 \\ 0 & r \neq s \end{cases} \tag{5.41}$$

Los coeficientes b_r se pueden encontrar multiplicando cada una de las ecuaciones de (5.40) por el valor apropiado de $T_s(x_j)(j = 0, 1, \ldots, n)$ y sumando las ecuaciones.

La propiedad de ortogonalidad hace que todos los términos de la parte derecha, excepto uno, sean cero. Esto conduce a

$$b_s = \frac{2}{n} \sum_{j=0}^{n}{}'' f(x_j) T_s(x_j) \tag{5.42}$$

Naturalmente, es de interés saber cómo hacer la conexión entre los coeficientes definidos por la expansión finita de Tchebyshev, dada por $\phi(x)$, y los coeficientes de la serie infinita. Si la expresión para serie infinita $f(x)$ se inserta en la ecuación (5.41), se obtiene

$$b_s = \frac{2}{n} \sum_{j=0}^{n}{}'' \sum_{k=0}^{\infty}{}'' a_k T_k(x_j) T_s(x_j)$$

Usando la definición de polinomios de Tchebyshev $T_k(x) = \cos k\theta$ y $x = \cos\theta$, se pueden usar las propiedades de los cosenos para mostrar que todos los términos $k = 2n(M \pm s)$ van a satisfacer la propiedad de ortogonalidad de (5.38), si M es un entero positivo $M \geq 1$. Esto se debe a que

$$\cos \frac{k\pi j}{n} = \cos \frac{k\pi s}{n}$$

para valores de k tales que

$$\frac{k\pi j}{n} = 2\pi M'' \pm \frac{k\pi s}{n} \quad \text{o} \quad k = \frac{2M''n}{j} \pm s$$

Debido a que k debe de ser entero, se restringen aquellos valores de M'' que son múltiplos de j, dando $k = 2Mn \pm s$. De esta manera, sólo estos valores de k conducirán a valores diferentes de cero en la parte derecha de la ecuación, y esto dará

$$b_s = a_s + a_{2n-s} + a_{2n+s} + a_{4n-s} + a_{4n+s} + \cdots$$

Si la serie tiene buenas propiedades de convergencia, los términos de grado alto serán pronto despreciables, y los b_s serán una buena aproximación de los coeficientes de la serie infinita. La sección 5.9.6 de este capítulo proporciona el código Matlab para hacer el ajuste de una función utilizando la técnica de Tchebyshev. Adicionalmente, la sección 5.9.7 proporciona el código Matlab para hacer un interpolador de Lagrange utilizando los puntos de ortogonalidad de Tchebyshev.

5.7.5 Evaluación de las series de Tchebyshev

La evaluación de polinomios se puede hacer usando un algoritmo de multiplicación recursiva (anidada). Entonces un posible camino de evaluación de la serie finita de Tchebyshev (5.39) será calcular los coefi-

www.FreeLibros.me

cientes de las distintas potencias de x, y usar este algoritmo. No obstante, hay un algoritmo que se puede usar directamente en las series de Tchebyshev, el cual es muy similar al algoritmo de la multiplicación recursiva. Este método es preferible ya que evita el reacomodo en x dentro de las potencias.

Así, por ejemplo, si se deja que $c_{n+1} = c_{n+2} = 0$ y se usa la relación de recurrencia:

$$c_r = 2xc_{n+1} - c_{n+2} + b_r, \qquad r = n, n-1, \ldots, 0,$$

el valor de la serie de Tchebyshev (5.39) en un punto x está dado por

$$\phi(x) = \frac{1}{2}(c_0 - c_2)$$

5.7.6 Otras propiedades de las series de Tchebyshev

La propiedad de Tchebyshev de oscilaciones iguales tiene una aplicación interesante cuando los puntos se eligen para usarse como base para la fórmula de interpolación. Se ha demostrado en la sección 5.2 (ecuación 5.5) que el error en la fórmula de interpolación usando los puntos $(x_0, \ldots, x_n)$ está dado por

$$(x - x_0)(x - x_1)\ldots(x - x_n)\frac{f^{(n+1)}(\zeta)}{(n+1)!}$$

donde ζ es algún punto en el intervalo de interpolación. Naturalmente, se requiere que este término de error sea lo más pequeño posible. Aunque poco se puede hacer para minimizar el término de la derivada, es ciertamente posible darle a los términos del producto la oscilación mínima eligiendo los puntos $x_i = (i = 0, 1, \ldots, n)$ para ser los ceros del polinomio de Tchebyshev $T_{n+1}(x)$.

Es también de interés el hecho de que los polinomios de Tchebyshev se usan en la fórmula de integración gaussiana. Éstos conducen a una fórmula que tiene coeficientes iguales que reducen ligeramente los requerimientos de cómputo. La presencia del factor de peso $(1+x^2)^{-1/2}$ usado en la definición de ortogonalidad de Tchebyshev facilita la posibilidad de evaluar numéricamente ciertas integrales con singularidades en el integrando.

EJEMPLO 5.8

Comprobar el teorema 5.2 partiendo de la construcción de un polinomio interpolador de Tchebyshev de 4 términos utilizando una función analítica dentro del intervalo [8, 12]. La función analítica es la siguiente:

$$f(x) = 2x - x^2 + x^3 \operatorname{sen}(x) e^{-\frac{x}{10}}$$

El polinomio interpolador de Tchebyshev de 4 términos tiene la siguiente estructura:

$$\varphi(x) = \sum_{k=0}^{3}{}'' b_k T_k(x) = \frac{b_0}{2}T_0(x) + b_1T_1(x) + b_2T_2(x) + \frac{b_3}{2}T_3(x)$$

Los polinomios de Tchebyshev son ortogonales dentro del rango $[-1, +1]$. Así, los 4 puntos necesarios para la construcción del polinomio deseado están determinados por

$$x_j = \cos[\pi j/n] \quad j = 0, 1, 2, 3 \qquad n = 3$$

Desarrollando la ecuación anterior, se tienen los siguientes puntos:

$$x_0 = \cos(0) = 1$$

$$x_1 = \cos(\pi/3) = 1/2$$

$$x_2 = \cos(2\pi/3) = -1/2$$

www.FreeLibros.me

$$x_3 = \cos(\pi) = -1$$

Adicionalmente, los primeros 4 polinomios de Tchebyshev son:

$$T_0(x) = 1$$

$$T_1(x) = x$$

$$T_2(x) = 2x^2 - 1$$

$$T_3(x) = 4x^3 - 3x$$

Para la realización de un polinomio interpolador de Tchebyshev en el rango [8, 12] es necesario transportar cada uno de los puntos ortogonales del intervalo [−1, +1] a este intervalo aplicando la siguiente ecuación:

$$z_i = \frac{1}{2}[(b-a)x_i + b + a]$$

Sustituyendo los límites del intervalo, es decir, $a = 8$ y $b = 12$, se obtiene

$$z_i = 2x_i + 10,$$

quedando los puntos de ortogonalidad trasladados de la siguiente manera:

$$z_0 = 2x_0 + 10 = 12$$

$$z_1 = 2x_1 + 10 = 11$$

$$z_2 = 2x_2 + 10 = 9$$

$$z_3 = 2x_3 + 10 = 8$$

Una vez que se tienen los puntos ortogonales en el intervalo en el cual se quiere hacer el polinomio interpolador, se evalúa la función $f(z) = 2z - z^2 + z^3 \operatorname{sen}(z) e^{-\frac{z}{10}}$ en esos puntos y los polinomios ortogonales de Tchebyshev en los cuatro puntos de ortogonalidad dentro del intervalo [−1, +1]. Los resultados se resumen en la tabla 5.9.

Tabla 5.9 Puntos de ortogonalidad transportados al intervalo [8, 12] y la función evaluada en esos puntos, y los primeros 4 polinomios de Tchebyshev evaluados en los puntos de ortogonalidad en [–1, +1].

Función analítica			Polinomios de Tchebyshev				
z	$f(z)$		x	$T_0(x)$	$T_1(x)$	$T_2(x)$	$T_3(x)$
$z_0 = 12$	−399.266671		1	1	1	1	1
$z_1 = 11$	−542.047073		1/2	1	1/2	−1/2	1
$z_2 = 9$	59.147501		−1/2	1	−1/2	−1/2	1
$z_3 = 8$	179.608225		−1	1	−1	1	−1

La formulación para los coeficientes b_s es (ecuación 5.42 con $n = 3$)

$$b_s = \frac{2}{3}\sum_{j=0}^{3}{}'' f(z_j) \cdot T_s(x_j)$$

www.FreeLibros.me

donde b_0 y b_3 tienen una función de peso de 1/2. Así se obtiene

$$b_0=\frac{2}{3}\sum_{j=0}^{3}{}'' f(z_j)\cdot T_0(x_j)=\frac{2}{3}\left[\frac{f(z_0)T_0(x_0)}{2}+f(z_1)T_0(x_1)+f(z_2)T_0(x_2)+\frac{f(z_3)T_0(x_3)}{2}\right]$$
$$=\frac{2}{3}\left[\frac{(-399)\cdot(1)}{2}+(-542)\cdot(1)+(59)\cdot(1)+\frac{(179)\cdot(1)}{2}\right]=-395.1525$$

$$b_1=\frac{2}{3}\sum_{j=0}^{3}{}'' f(z_j)\cdot T_1(x_j)=\frac{2}{3}\left[\frac{f(z_0)T_1(x_0)}{2}+f(z_1)T_1(x_1)+f(z_2)T_1(x_2)+\frac{f(z_3)T_1(x_3)}{2}\right]$$
$$=\frac{2}{3}\left[\frac{(-399)\cdot(1)}{2}+(-542)\cdot(1/2)+(59)\cdot(-1/2)+\frac{(179)\cdot(-1)}{2}\right]=-393.3565$$

$$b_2=\frac{2}{3}\sum_{j=0}^{3}{}'' f(z_j)\cdot T_2(x_j)=\frac{2}{3}\left[\frac{f(z_0)T_2(x_0)}{2}+f(z_1)T_2(x_1)+f(z_2)T_2(x_2)+\frac{f(z_3)T_2(x_3)}{2}\right]$$
$$=\frac{2}{3}\left[\frac{(-399)\cdot(1)}{2}+(-542)\cdot(-1/2)+(59)\cdot(-1/2)+\frac{(179)\cdot(1)}{2}\right]=87.7470$$

$$b_3=\frac{2}{3}\sum_{j=0}^{3}{}'' f(z_j)\cdot T_3(x_j)=\frac{2}{3}\left[\frac{f(z_0)T_3(x_0)}{2}+f(z_1)T_3(x_1)+f(z_2)T_3(x_2)+\frac{f(z_3)T_3(x_3)}{2}\right]$$
$$=\frac{2}{3}\left[\frac{(-399)\cdot(1)}{2}+(-542)\cdot(-1)+(59)\cdot(1)+\frac{(179)\cdot(-1)}{2}\right]=207.8381$$

Por tanto, el polinomio de Tchebyshev de 4 términos, para aproximar la función $f(z)=2z-z^2+z^3\text{sen}(z)e^{-\frac{z}{10}}$ en el rango [8, 12], es el siguiente:

$$\varphi(x)=\sum_{k=0}^{3}{}'' b_kT_k(x)=\frac{b_0}{2}T_0(x)+b_1T_1(x)+b_2T_2(x)+\frac{b_3}{2}T_3(x)$$

La aproximación de Tchebyshev está determinada por la expresión

$$f(x)=\sum_{k=0}^{3}{}' a_kT_k(x)=\frac{a_0}{2}T_0(x)+a_1T_1(x)+a_2T_2(x)+a_3T_3(x)$$

Relacionando término a término las dos funciones anteriores, se tiene que $[f(x_j)=\varphi(x_j)]$, de tal forma que los coeficientes buscados son los siguientes:

$$\frac{a_0}{2}=\frac{b_0}{2}=\frac{-395}{2}$$
$$a_1=b_1=-393$$
$$a_2=b_2=87$$
$$a_3=\frac{b_3}{2}=\frac{207}{2}=103$$

El polinomio resultante es un interpolador de Tchebyshev de 4 términos, el cual es una aproximación de la función $f(x)=2x-x^2+x^3\ \text{sen}(x)e^{-\frac{x}{10}}$ de la forma

$$f(x)=\frac{a_0}{2}(1)+a_1(x)+a_2(2x^2-1)+a_3(4x^3-3x)$$

www.FreeLibros.me

Sustituyendo valores queda finalmente

$$f(x) = 415.6762\,x^3 + 175.4941\,x^2 - 705.1136\,x - 285.3233$$

En resumen, la forma de utilizar este polinomio interpolador de Tchebyshev es la siguiente:

1. Se toman puntos en el intervalo [−1, +1] y se evalúan con este polinomio.
2. El punto se transporta al intervalo en el cual se calcula la función aproximada [8, 12].
3. El valor obtenido en el polinomio es válido en el punto transportado al intervalo [8, 12].

Con la finalidad de comprobar el teorema 5.2 que enuncia que dos polinomios de tercer orden construidos a partir de los mismos 4 puntos son equivalentes dentro del intervalo, con los puntos de ortogonalidad arrojados por el método anterior para construir polinomios interpoladores de Tchebyshev, se construye un polinomio interpolador de Lagrange. Estos puntos y el valor de la función se muestran en la tabla 5.10.

Tabla 5.10 Puntos ortogonales de Tchebyshev para el interpolador de Lagrange y la función en esos puntos.

x	$x_0 = 8$	$x_1 = 9$	$x_2 = 11$	$x_3 = 12$
$f(x)$	$f(x_0) = 179.6082$	$f(x_1) = 59.1475$	$f(x_2) = -542.0470$	$f(x_3) = -399.2667$

Utilizando los valores de la tabla 5.10, el polinomio interpolador de Lagrange queda de la siguiente manera:

$$f(x) = 51.9595\,x^3 - 1514.9121\,x^2 + 14357.8291\,x - 44331.9242$$

La figura 5.12 muestra las gráficas tanto de la función analítica como de los polinomios interpoladores de Tchebyshev y Lagrange. Por otro lado, la figura 5.13 muestra el error porcentual de ambas aproximaciones polinómicas cuando se usan los puntos base dados por el método de Tchebyshev. Se hace notar, por el análisis de esta figura, que el error en los puntos base, es decir en 8, 9, 11 y 12, es cero. Igualmente se hace notar que estos puntos no están igualmente espaciados. La figura 5.13 muestra que, efectivamente, el polinomio interpolador de Tchebyshev y el de Lagrange son equivalentes dentro del intervalo [8, 12].

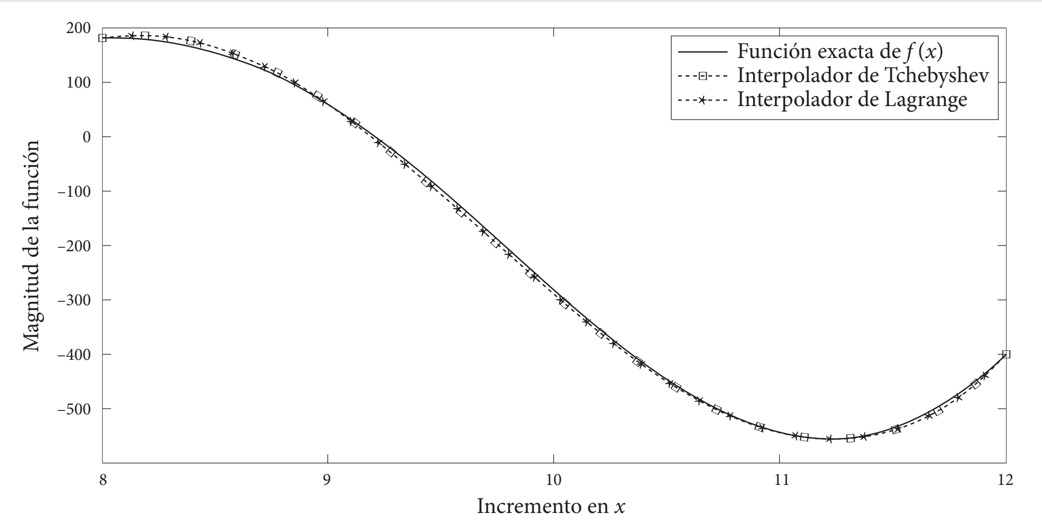


Figura 5.12 Interpoladores de Tchebyshev y de Lagrange con puntos base ortogonales.

www.FreeLibros.me

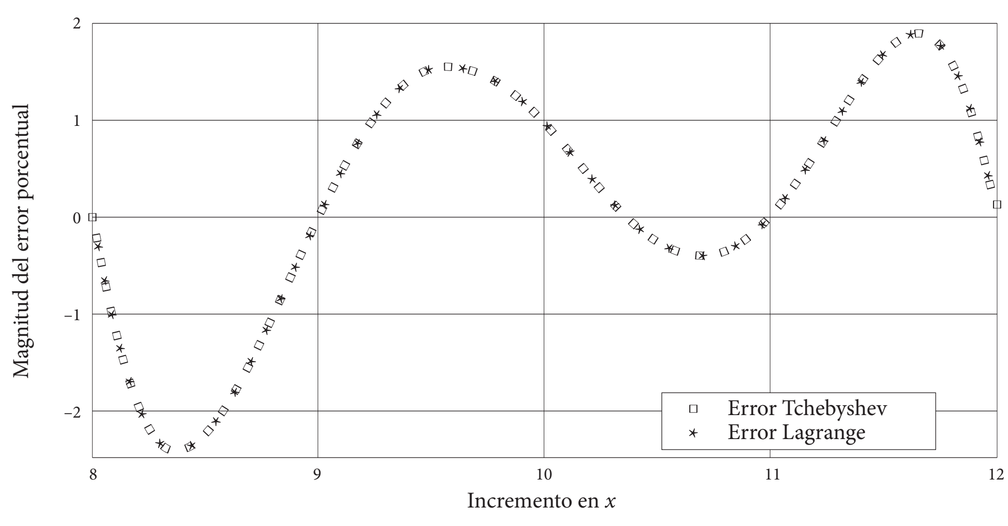


Figura 5.13 Errores porcentuales en el intervalo [8, 12] dado por los dos polinomios interpoladores.

El ejemplo anterior muestra cómo la construcción de un polinomio interpolador de Tchebyshev es mucho más complicada que la construcción de un polinomio interpolador de Lagrange. Asimismo, muestra que basta utilizar los puntos de ortogonalidad discretos determinados por el proceso de Tchebyshev para obtener un error determinado por el método de Tchebyshev. Con esto se demuestra el teorema 5.2, y que el error depende de los puntos utilizados para construir el polinomio interpolador y no del método que se siga.

Adicionalmente, la evaluación del polinomio interpolador de Tchebyshev resultante es mucho más complicado, pues primero se evalúa dentro del intervalo $[-1, +1]$, y después cada punto se transporta al intervalo de interés $[a, b]$. En cambio, el polinomio de Lagrange se evalúa directamente dentro del intervalo de interés.

5.8 Comparación de métodos

El concepto de interpolación esta íntimamente ligado al concepto de ajuste de un grupo de datos por una curva. Esencialmente, la única diferencia consiste en que la construcción de un polinomio interpolador hace el error cero en los puntos dados como información, y el orden del polinomio será siempre de orden $n - 1$, donde n es el número de datos. En cambio, el concepto de ajustar una curva, en forma adicional, tiene dos variantes al de un polinomio interpolador. La primera es ajustar una curva por una serie ortogonal como lo realiza la técnica de Fourier, y la segunda variante se refiere a que se puede hacer el ajuste de un grupo de datos por un polinomio de bajo orden, inclusive cero. Por supuesto, si por esta filosofía de ajuste se construye un polinomio de orden $n-1$, como ya se demostró y lo enuncia el Teorema 5.2, el resultado será equivalente a la construcción de un polinomio interpolador por cualquier técnica.

Todos los polinomios en realidad, si parten de la misma información y son del mismo orden, son equivalentes dentro del intervalo donde están los datos. Por tanto, se vuelve lógico adoptar el de construcción más simple.

Se debe tener en cuenta que el error depende de la posición de los datos. Aunque parezca lógico que una distribución homogénea debe ser la que dé el menor error, se demostró que, en realidad, los datos con una distribución que tengan la característica de ortogonalidad discreta dan el menor error: pero las técnicas de construcción de polinomios interpoladores con esta teoría son muy complejas. Por la razón anterior, si se tiene control en la toma de datos, se hace una combinación de técnicas. Por un lado, se

www.FreeLibros.me

utiliza la distribución que cumpla con la ortogonalidad discreta para minimizar el error y, partiendo de estos puntos, se utiliza una metodología de construcción simple como la de Lagrange. Finalmente, el Teorema 5.2 garantiza que el error se mantiene dentro del intervalo independientemente de la técnica de construcción del polinomio interpolador.

5.9 Programas desarrollados en Matlab

Esta sección proporciona los códigos de los programas desarrollados en Matlab para todos los ejercicios propuestos. A continuación se enlistan todos ellos:

5.9.1 Matriz de Vandermonde
5.9.2 Interpolación de Lagrange
5.9.3 Diferencias divididas de Newton
5.9.4 Mínimos cuadrados
5.9.5 Ajuste por la transformada de Fourier
5.9.6 Ajuste por polinomios de Tchebyshev
5.9.7 Utilizar los puntos de Tchebyshev para un interpolador de Lagrange

5.9.1 Matriz de Vandermonde

El método de la matriz de Vandermonde calcula los coeficientes de un polinomio interpolador partiendo de una tabla de datos. La única limitante de este método es el cálculo de la matriz inversa, matriz de Vandermonde, debido a que esta matriz es de las llamadas matrices mal condicionadas.

Programa principal del método de Vandermonde

```
% Construcción de una función de orden "n" para interpolar un grupo de datos
% utilizando el método de la matriz de Vandermonde.
clear all
clc
% Vector de la variable independiente
x = [1; 3; 6; 8; 9; 11; 13];
% Valor de la función en los puntos "x".
fx = [16; 31; 12; 21; 9; 42; 19];
% Número de muestras
N = length(x);
% Rutina para crear la matriz de coeficientes
V = [];
for k = 1:N
 V = [V x.^(k-1)];
end
A = inv(V)*fx;        % Coeficientes calculados por el método de la matriz de
                      % Vandermonde.
A = A.';              % Para invertir el orden de los coeficientes se necesita un
                      % vector renglón.
A = fliplr(A);        % Se invierte el orden de los coeficientes para utilizar la
                      % función polyval.
xp = (1:0.01:13);     % Vector a evaluar en puntos entre el límite inferior y superior.
Fxp = polyval(A,xp); % Evaluación de la función en todos los puntos propuestos.
% Gráficas de los puntos iniciales y el polinomio interpolador
plot(x,fx,'o',xp,Fxp,'r'); legend('Puntos iniciales','Polinomio interpolador')
```

5.9.2 Interpolación de Lagrange

El método de interpolación de Lagrange crea un polinomio de orden $n-1$, donde n es el número de datos, a partir de una función discretizada o de un grupo de datos. La función polinómica que crea esta técnica es continua y puede calcular todos los puntos dentro del intervalo. En los puntos base, el error es cero y,

www.FreeLibros.me

fuera de ellos, si se tiene un punto de referencia se puede calcular el error si no se espera un error simplemente acotado.

Programa principal de la interpolación de Lagrange

```
% Función para crear un interpolador de Lagrange a partir de los siguientes datos:
%     x .- vector discreto de la variable independiente
%     fx .- valor de la función en los puntos x
clear all
clc
% Vector de la variable independiente
x = [2; 4; 6; 8; 10; 12; 14];
% Valor de la función en los puntos "x".
fx = [5; 9; 2; 1; 6; 4; 9];
% Número de muestras
N = length(x);
% Rutina para calcular los coeficientes.
for k=1:N
    for m=1:N
        if k==m
            ind = 1;                 % Índice para omitir el dato cuando k=m.
        else
            FN(k,m-ind) = x(m);      % Factores del numerador (raíces del polinomio).
            FD(k,m-ind) = x(k)-x(m); % Factores del denominador.
        end
    end
    ind = 0;                         % Se pone en cero para sólo omitir el caso k=m.
end
% Cálculo de los coeficientes del numerador de cada polinomio.
for k = 1:N
    CN(k,:) = poly(FN(k,:));
end
Fac = prod(FD,2);   % Multiplica los factores de cada denominador.
% Rutina para multiplicar cada polinomio por el factor correspondiente.
for k = 1:N
    Multi(k,:) = CN(k,:)*fx(k)/Fac(k);
end
Coef = sum(Multi);        % Calcula los coeficientes del polinomio interpolador.
xp  = (2:0.01:14);        % Vector por evaluar en puntos entre el límite inferior y
                          % superior.
Fxp = polyval(Coef,xp);   % Evaluación de la función en todos los puntos propuestos.
% Gráficas de los puntos iniciales y el polinomio interpolador.
plot(x,fx,'o',xp,Fxp,'r'); legend('Puntos iniciales','Polinomio interpolador')
```

5.9.3 Método de diferencias divididas de Newton

El método de diferencias dividas de Newton toma un grupo de datos y crea un interpolador de la siguiente forma: con los primeros dos datos crea una diferencia (derivada numérica), con el segundo y tercer datos crea otra diferencia y así sucesivamente. Con las primeras dos diferencias crea a su vez otra diferencia y así sucesivamente. Se sigue con este proceso hasta que se tiene el número de todas las diferencias que se puede crear con el grupo de datos. Dentro de estos datos se encuentran los coeficientes que acompañan a la formulación de diferencias divididas de Newton.

Programa principal del método de diferencias divididas de Newton

```
% Función para crear un interpolador utilizando el método de diferencias divididas de
% Newton, a partir de los siguientes datos:
%     x .- vector discreto de la variable independiente
%     fx .- valor de la función en los puntos x
```

www.FreeLibros.me

```
clear all
clc
% Vector de la variable independiente.
x = [1; 3; 6; 9; 12; 15; 18];
% Valor de la función en los puntos "x".
fx = [1; 2; 1; 3; 1; 4; 1];
% Número de muestras.
N = length(x);
% Orden de la máxima diferencia dividida de Newton.
P = N-1;
% Aparta espacio para el almacenamiento de x, fx y todas las diferencias divididas de
% Newton.
Tb = zeros(N,P);
Tb = [x fx Tb];
% Rutina para crear la tabla columna por columna.
for k=1:P
    for m=1:N-k
        Num = Tb(m+1,k+1) - Tb(m,k+1);   % Diferencia en el numerador.
        Den = Tb(m+k,1) - Tb(m,1);       % Diferencia del denominador.
        Tb(m,k+2) = Num/Den;             % Coeficiente resultante de la división.
    end
end
% Los coeficientes quedan en el primer renglón de la columna 2 a la P+2, por tanto:
  Newton = Tb(1,2:P+2);
% Rutina que crea la matriz que contiene los coeficientes resultantes de multiplicar
% cada término de la tabla de diferencias divididas de Newton por el polinomio de
% grado k que lo acompaña.
for k=1:N
    orden = x(1:k-1);              % Orden del polinomio que acompaña al coeficiente
                                   % de Newton.
    Pol = Newton(k)*poly(orden);   % Obtiene los coeficientes de un polinomio de
                                   % orden k-1.
    Coef(k,N-k+1:N) =  Pol;        % Acomoda los coeficientes en la matriz Coef.
end
% Suma todos los términos del mismo orden para crear el polinomio interpolador.
Pol = sum(Coef);
xp  = (1:0.01:18);           % Vector a evaluar en puntos entre el límite inferior y
                             % superior.
Fxp = polyval(Pol,xp);       % Evaluación de la función en todos los puntos
                             % propuestos.
% Gráficas de los puntos iniciales y el polinomio interpolador.
plot(x,fx,'o',xp,Fxp,'r');   legend('Puntos iniciales','Polinomio interpolador')
```

5.9.4 Método de mínimos cuadrados

El método de mínimos cuadrados se utiliza para ajustar un grupo de datos con una función polinómica desde orden cero hasta $n-1$. Es decir, con esta técnica se puede ajustar un grupo de veinte datos con un polinomio de bajo orden (inclusive cero) o con un polinomio de alto orden como sería el orden máximo (19) para este grupo de datos.

Programa principal del método de mínimos cuadrados

```
% Función para crear un polinomio de grado n que ajusta un grupo de datos utilizando
% la técnica de mínimos cuadrados, a partir de los siguientes datos:
%      x .- vector discreto de la variable independiente
%      fx .- valor de la función en los puntos x
clear all
clc
format long g
% Vector de la variable independiente.
x  = [1.1   1.7    2.9    3.7    4.5];
% Valor de la función en los puntos "x".
```

www.FreeLibros.me

```
fx = [3.41  5.17  23.46  36.45  40.86];
% Número de muestras.
M = length(x);
% Orden del polinomio que se desea crear.
P = M-4;
% Orden de la matriz de coeficientes.
N = P+1;
% Rutina para crear todos los elementos de la matriz de coeficientes.
for k = 1:2*P+1
    a(k,1) = sum(x.^(k-1));
end
% Rutina para crear la matriz de coeficientes.
for k=1:N
    A(:,k) = a(k:N+k-1);
end
% Rutina para crear los valores de YiXi.
for k=1:N
    Y(k,1) = sum(fx.*x.^(k-1));
end
% Cálculo de los coeficientes del polinomio que ajusta el grupo de datos.
ck = inv(A)* Y;
% Se invierte el orden de los coeficientes para utilizar la función polyval para
% evaluarlo.
ck = fliplr((ck.'));
xp  = (1.1:0.01:4.5);   % Vector a evaluar en puntos entre el límite inferior y
                        % superior.
Fxp = polyval(ck,xp);   % Evaluación de la función en todos los puntos propuestos.
% Gráficas de los puntos iniciales y el polinomio interpolador.
plot(x,fx,'o',xp,Fxp,'r'); legend('Puntos iniciales','Ajuste por mínimos cuadrados')
```

5.9.5 Ajuste utilizando la transformada discreta de Fourier

La transformada discreta de Fourier se puede utilizar para hacer el ajuste de un grupo de datos mediante un conjunto finito de funciones ortogonales. El grupo de funciones resultante se evalúa en los puntos discretos y reproduce la función discreta. La ventaja de tener un conjunto de funciones analíticas en vez de un grupo de datos es, por supuesto, la versatilidad de la manipulación algebraica, sabiendo de antemano que son válidas sólo en los puntos definidos por la variable independiente.

Programa principal del ajuste con la transformada discreta de Fourier

```
% Función para hacer un ajuste discreto por la técnica de Fourier, a partir de los
% siguientes datos:
%     tp .- vector discreto de la variable independiente.
%     ft .- valor de la función en los puntos tp.
clear all
clc
format long g
Ti = 0;                       % Tiempo inicial.
Tf = 7.5;                     % Tiempo final.
n  = 7;                       % Potencia para sacar el número de muestras.
Ns = 2^n;                     % Número de muestras en potencias de 2.
h  = (Tf-Ti)/(Ns-1);          % Incremento.
tp = [Ti:h:Tf];               % Vector de tiempos para calcular la función discreta.
ft = exp(-tp/2).*sin(2*tp);   % Calcula una función en tiempo discreto.
Fs = fft(ft);                 % Transformada de Fourier para calcular las amplitudes.
Dg = (2*pi/Ns);               % Incremento angular de cada muestra.
Arg(Ns,Ns) = zeros;           % Inicializa la matriz de argumentos de la serie
                              % discreta.
% Cálculo de todos los argumentos de la serie discreta.
for k = 1:Ns-1
    Arg(k+1,:) = [0:k*Dg:k*2*pi-k*Dg];
```

www.FreeLibros.me

```
end
% Cálculo de todas las funciones de la serie.
for k = 1:Ns
    CIS(k,:) = (1/Ns).*Fs(k).*exp(j*Arg(k,:));
end
% Suma muestra a muestra la parte real de la serie.
Ft = sum (real(CIS),1);
% Gráfica de la función tabular y la función resultante de la TDF.
plot(tp,ft,'o',tp,Ft,'.-',tp,Ft1,'g'), grid
legend('Función tabular de t vs ft','Función resultante de la TDF')
xlabel('Número de muestras'), ylabel('Amplitud de la función')
```

5.9.6 Ajuste de Tchebyshev

La técnica de ajustar un grupo de datos mediante polinomios de Tchebyshev se usa esencialmente cuando se tiene control sobre la toma de datos; es decir, se pueden tomar los datos donde se requieran. De esta forma, se utilizan los puntos ortogonales que dan los polinomios de Tchebyshev y así se garantiza que el error tiene máximos y mínimos acotados.

Programa principal del ajuste de Tchebyshev

```
% Función para crear un interpolador utilizando el método de Tchebyshev cuando se
% tiene un intervalo de ajuste y una función analítica.
%      [a,b] .- intervalo de ajuste
%      fx .- función analítica
clear all
clc
format long g
% Intervalo de interés.
a = 8;    % Límite inferior del intervalo.
b = 12;   % Límite superior del intervalo.
% Número de términos o puntos de ortogonalidad que se quieren.
N = 17;
% Orden del polinomio.
P = N-1;
% Se crea en forma inicial una celda de N espacios para almacenar el polinomio de
% Tchebyshev.
T = cell(1,N);
% Inicia los dos primeros polinomios de Tchebyshev y los acomoda en su respectiva
% celda.
T(1:2) = { [1], [1 0] };
% Relación de recurrencia para calcular los coeficientes de Tchebyshev, en este caso
% los almacena en celdas de dimensión adecuada de acuerdo al orden del polinomio.
for k = 2:N-1
    T{k+1} = [2*T{k} 0] - [0 0 T{k-1}];
end
% Coeficientes de los polinomios a utilizar en el ajuste.
Tchebyshev(N,N) = zeros;
for k=1:N
    Tchebyshev(k,1:k)=T{k};
end
Tshe = Tchebyshev.';
% Cálculo de los puntos de ortogonalidad para el ajuste de Tchebyshev.
x = (cos(((N-1:-1:0)* pi)/(N-1))).';
% Cambio de los puntos de ortogonalidad al intervalo [a=8 b=12] con la fórmula
% z=(1/2)[(b-a)x + b + a].
  z = (1/2).*((b-a).* x + b + a);
% Evaluación de la función en los puntos de ortogonalidad trasladados.
fz = 2.*z - z.^2 + z.^3.*sin(z).*exp(-z./10);
% Cálculo de la matriz de datos Shev que contiene
% |  z  |  fz  |  x  |  To(x)  |  T1(x)  |  ....  |  Tn(x)  |
Shev = [];               % Inicia la variable en vacío [].
```

www.FreeLibros.me

```
Shev = [Shev z fz x];   % Sobre la matriz de datos por z, fz y x.
Pe = [];                % Inicia la evaluación de los polinomios de Tchebyshev en
                        % vacío [].
% Ciclo para evaluar numéricamente los polinomios de Tchebyshev.
for l = 1:N
    Ren(N,N)=zeros;     % Esta variable almacena la variable x elevada a todas las
                        % potencias, desde o hasta N-1.
    for k=1:N
        for m=k:-1:1
            Ren(m,k)=x(l).^(k-m);
        end
    end
    M = Tshe.*Ren;      % Se hace la multiplicación de los coeficientes por la
                        % variable x correspondiente.
    C = (sum(M));       % Se suma el resultado de cada término de los polinomios para
                        % tener su evaluación numérica.
    Pe = [Pe; C];       % El resultado de cada polinomio evaluado se guarda en una
                        % matriz Pe.
end
Shev = [Shev Pe];       % Finalmente se le agrega a la matriz de datos Shev.
% Rutina para calcular los coeficientes B.
B = (2/(N-1)).* Shev(1,2)*Shev(1,4:3+N)/2;
for k=2:N-1
    B = B + (2/(N-1)) .* Shev(k,2).*Shev(k,4:3+N);
end
B = B + (2/(N-1)) .* Shev(N,2)*Shev(N,4:3+N)/2;
% Rutina para el cálculo de los coeficientes A.
A(1)=B(1)/2;
A(2:N-1) = B(2:N-1);
A(N)= B(N)/2;
% Rutina para calcular los coeficientes que acompañan los polinomios de Tchebyshev.
for k=1:N
    CO(:,k) = A(k)*Tshe(:,k);
end
% Rutina para calcular los coeficientes del polinomio interpolador.
for k=1:N
    Coef(k) = sum(diag(CO,k-1));
end
% Polinomio interpolador de Tchebyshev evaluado en el intervalo [-1, +1].
w = -1:0.01:1;
fx = 0;
for k=1:N
    fx = fx + Coef(k)*w.^(k-1);
end
% Se transportan los puntos del intervalo [-1,+1] al intervalo de interés [a,b].
q  = (1/2).*((b-a).*w + b + a);
% Evaluación de la función analítica (como punto de comparación) en los puntos de
% ortogonalidad trasladados.
fq = 2.*q - q.^2 + q.^3.*sin(q).*exp(-q./10);
% Gráfica de los puntos utilizados para hacer el polinomio interpolador, de la
% función analítica evaluada y del polinomio interpolador de Tchebyshev.
plot(z,fz,'o',q,fx,'r',q,fq,'--');
legend('Puntos utilizados para crear el interpolador','Interpolador de
Tchebyshev','Función analítica')
```

5.9.7 Interpolador de Lagrange que utiliza los puntos de Tchebyshev

La técnica de ajustar un grupo de datos mediante polinomios de Tchebyshev se usa esencialmente cuando se tiene control sobre la toma de datos; es decir, se pueden tomar los datos donde se requieran. De esta forma se utilizan los puntos ortogonales que dan los polinomios de Tchebyshev y de esta forma se garantiza que el error tiene máximos y mínimos acotados.

www.FreeLibros.me

Programa principal del interpolador de Lagrange-Tchebyshev

```
% Función para crear un interpolador utilizando el método de Lagrange partiendo de
% los puntos resultantes de los polinomios de Tchebyshev cuando se tiene un intervalo
% de ajuste y una función analítica.
%   [a,b] .- intervalo de ajuste.
%   fx .-    función analítica.
%   N .-     Número de términos que se quieren utilizar.
clear all
clc
a=1;                                  % Límite inferior del intervalo de ajuste.
b=5;                                  % Límite superior del intervalo de ajuste.
N = 4;                                % Número de términos o puntos de ortogonalidad
                                      % necesarios.
P = N-1;                              % Orden del polinomio.
x = cos(((N-1:-1:0)* pi)/(N-1));      % Cálculo de los puntos de ortogonalidad de
                                      % Tchebyshev.
% Cambio de los puntos de ortogonalidad al intervalo [a b] con la fórmula
% Zj=(1/2)[(b-a)Xj + b + a].
z = (1/2).*((b-a).*x + b + a);              % Puntos para la interpolación
fz = exp(-z).*log(z) - z.^3 + 5*z.^2 + 2*z; % Función analítica evaluada en los
                                            % puntos a interpolar
% Rutina para calcular los coeficientes.
for k=1:N
    for m=1:N
        if k==m
            ind = 1;                        % Índice para omitir el dato cuando
                                            % k=m.
        else
            FN(k,m-ind) = z(m);             % Factores del numerador (raíces del
                                            % polinomio).
            FD(k,m-ind) = z(k)-z(m);        % Factores del denominador.
        end
    end
    ind = 0;                                % Se pone en cero para sólo omitir el
                                            % caso k=m.
end
% Cálculo de los coeficientes de cada polinomio del numerador.
for k = 1:N
    CN(k,:) = poly(FN(k,:));
end
Fac = prod(FD,2);   % Multiplica los factores de cada denominador.
% Rutina para multiplicar cada polinomio por el factor correspondiente.
for k = 1:N
    Multi(k,:) = CN(k,:)*fz(k)/Fac(k);
end
Coef = sum(Multi);
xp   = (1:0.01:5);                                  % Vector a evaluar en puntos
                                                    % entre el límite inferior y
                                                    % superior.
Fxp = polyval(Coef,xp);                             % Evaluación de la función en
                                                    % todos los puntos propuestos.
fza = exp(-xp).*log(xp) - xp.^3 + 5*xp.^2 + 2*xp;   % Función analítica evaluada en
                                                    % los puntos a interpolar.
% Gráficas de los puntos iniciales y el polinomio interpolador
plot(z,fz,'o',xp,Fxp,'r',xp,fza,'k'); legend('Puntos iniciales','Polinomio
interpolador','Función analítica')
```

www.FreeLibros.me

Problemas propuestos

5.10.1. Construya un interpolador utilizando la técnica de Lagrange para encontrar los valores de la función en los puntos 1.2, 1.9 y 2.1, dados por la siguiente tabla:

X	1.1	1.7	2.9	3.7	**4.5**
$f(x)$	3.41	5.17	23.46	36.45	40.86

5.10.2 Use interpolación de Lagrange de tercer orden para encontrar los valores de la función en los puntos 2.0 y 4.0, dados por la siguiente tabla de valores:

x	1.4	2.6	3.2	**4.5**
$f(x)$	0.725	0.548	0.423	0.173

5.10.3 Utilizando la técnica de Lagrange, construya un interpolador que pasa por los puntos dados en la siguiente tabla:

x	−6	0	4	7
$F(x)$	13.56	−7.65	−1.24	87.97

5.10.4 Construya un polinomio interpolador de Lagrange de segundo orden, partiendo de los siguientes datos:

x	2	5	8
$F(x)$	12.4	5.7	17.1

5.10.5 Utilizando la técnica de Lagrange, construir un interpolador que pasa por los puntos dados en la siguiente tabla:

x	12	18	19	60
$F(x)$	1234	1100	1096	1979

5.10.6 Construya un polinomio interpolador de Lagrange de segundo orden, partiendo de los siguientes datos:

X	200	201	202	203	204	205
$f(x)$	2	1	2	3	2	2

5.10.7 Construya un polinomio interpolador de Lagrange de segundo orden, partiendo de los siguientes datos:

X	0.01	0.03	0.06	0.1	0.15	0.21	0.27	0.34
$f(x)$	1	0.9	0.8	0.7	0.8	0.9	1	3

5.10.8 Construya un polinomio interpolador que pasa por los puntos (1, 6), (3, 27), (5, 62), (7, 134) y (9, 46) con la formulación de diferencias divididas de Newton.

www.FreeLibros.me

5.10.9 Construya la tabla de diferencias divididas de Newton, partiendo de los puntos (3, 27), (5, 12), (7, 1), (9, 18), (11, 32) y (13, 49).

5.10.10 Con la formulación de diferencias divididas de Newton construya un polinomio interpolador que pasa por los puntos dados en la siguiente tabla:

X	1	2	3	4	5
f(*x*)	9	5	7	13	26

5.10.11 Con la formulación de diferencias divididas de Newton construya un polinomio interpolador que pasa por los puntos dados en la siguiente tabla:

X	1	2	3	5	7	11	13	17	18
f(*x*)	10	8	6	4	2	6	10	30	40

5.10.12 Con la formulación de diferencias divididas de Newton construya un polinomio interpolador que pasa por los puntos dados en la siguiente tabla:

X	1	4	6	10	**14**	22	26
f(*x*)	2	3	4	5	6	7	4

5.10.13 Con la formulación de diferencias divididas de Newton construya un polinomio interpolador que pasa por los puntos dados en la siguiente tabla:

X	5	6	7	8
f(*x*)	1	3	14	15

5.10.14 Con la formulación de diferencias divididas de Newton construya un polinomio interpolador que pasa por los puntos dados en la siguiente tabla:

X	8	16	24
f(*x*)	5	6	4

5.10.15 Seleccione los siete puntos óptimos desde el punto de vista de ortogonalidad discreta, para hacer las mediciones y construya un polinomio interpolador de Lagrange. Se tiene que el intervalo de interés de [0, 100]; si se toman como las mediciones en esos puntos ortogonales los valores de [13, 29, 58, 7, 43, 99, 0] respectivamente, construya un polinomio interpolador de sexto orden.

5.10.16 Si se sabe que los valores [0.25, 1.15, 0.97, 1.03] son cuatro mediciones que vienen de puntos de ortogonalidad discreta en el intervalo [36, 39], construya un polinomio interpolador de Lagrange de orden 3.

5.10.17 Si se sabe que los valores [12.45; 24.54; 2.43; 32.9; 92.34; 18.43; 27.16; 99.71] son ocho mediciones que vienen de puntos de ortogonalidad discreta en el intervalo [33, 48], construya un polinomio interpolador de Lagrange de orden 7.

5.10.18 Calcule los nueve puntos que cumplen con el concepto de ortogonalidad discreta dentro del intervalo [4.257, 13.739].

www.FreeLibros.me

5.10.19 Usando la siguiente tabla de datos, haga una aproximación de tercer orden (un polinomio cúbico) por el método de mínimos cuadrados.

x	–9	–5	–1	0	5	7	13
$f(x)$	–76	–52	–17	–4	18	35	44

Un polinomio cúbico tiene la forma $P_3(x) = a_0 + a_1x + a_2x^2 + a_3x^3$.

5.10.20 Construya un polinomio interpolador de orden 4 por el método de mínimos cuadrados, si se tienen los siguientes datos:

x	–3	–1	0	2	6	9	14	18	27
$f(x)$	–9	–5	–1	3	7	15	21	37	59

5.10.21 Utilizando la técnica de mínimos cuadrados, construya un polinomio interpolador de orden tres si se tienen los siguientes datos:

x	1	3	4	7	12	18	24	29	31	37	43	51	78
$f(x)$	9	4	1	0	–7	–9	–12	–17	–21	–24	–25	–32	–54

5.10.22 Utilizando la técnica de mínimos cuadrados, construya polinomios interpoladores de orden 2 y 3 para aproximar el grupo de datos dado por:

x	2	7	11	15	19	23	27
$f(x)$	7	6	5	4	3	2	1

Determine el error de cada polinomio, y póngalos en una gráfica junto con el grupo de datos para verificar si de manera visual se nota cuál tiene mayor error.

5.10.23 Utilizando la técnica de mínimos cuadrados, construya un polinomio interpolador de orden 2 si se tienen los siguientes datos:

x	1	3	4	7	11	13	17	19	23	27	31	37	39
$f(x)$	29	30	28	27	26	22	25	25	26	25	28	29	30

5.10.24 Utilizando la técnica de mínimos cuadrados, construya un polinomio interpolador de orden dos si se tienen los siguientes datos:

x	2	5	12
$f(x)$	2	1	3

5.10.25 Utilizando la técnica de mínimos cuadrados, construya un polinomio interpolador de orden seis si se tienen los siguientes datos:

x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$f(x)$	25	34	42	58	60	62	65	63	67	75	80	79	68	57	45	46

5.10.26 Utilizando la transformada discreta de Fourier, aproxime la curva dada por la función $f(t) = 20e^{-t^{1.4}}$, dentro del intervalo [0, 3], con un paso de discretización de $\Delta t = 0.2$.

5.10.27 Utilizando la transformada discreta de Fourier, aproxime la curva dada por la función $f(t) = \text{sen}(377t^{2.3})e^{-t/7}$, dentro del intervalo [0, 0.2], con un paso de discretización de $\Delta t = 0.2/31$.

www.FreeLibros.me

5.10.28 Utilizando la transformada discreta de Fourier, aproxime la curva dada por la función $f(t)=t^{2.37}e^{-t^{1.22}}$, dentro del intervalo [0, 7], con un paso de discretización de $\Delta t=7/31$.

5.10.29 Utilizando la transformada discreta de Fourier, haga la descomposición en la base ortogonal de la función tabulada dada por

t	0	0.2	0.4	0.6	0.8	1.0	1.2	1.4
$f(t)$	1.12	1.16	1.45	1.78	1.97	1.56	1.34	1.21

5.10.30 Utilizando la transformada discreta de Fourier, haga la descomposición en la base ortogonal de la función tabulada dada por

t	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5
$f(t)$	3.45	5.35	6.61	3.45	6.95	5.25	6.32	4.28	4.25	2.54	6.35	3.98	6.45	3.54	4.78	6.56

5.10.31 Dados los polinomios de Tchebyshev $T_0(x)=1$, $T_1(x)=x$ y la relación de recurrencia $T_{r+1}(x)+T_{r-1}(x)=2xT_r(x)$, $r=1,2,\ldots$, encuentre $T_2(x)$, $T_3(x)$ y $T_4(x)$.

5.10.32 Encuentre una expansión de cuatro términos de una serie finita de Tchebyshev que aproxime la función e^x en el intervalo [−1, +1].

5.10.33 Construya un interpolador de Tchebyshev de orden cinco para la función $\log(t^{0.5})e^{-0.01t}$ en el intervalo [3, 18].

5.10.34 Utilizando los polinomios de Tchebyshev, construya un polinomio interpolador de Tchebyshev de orden seis para la función $t^2 \tan 2t$ en el intervalo [−7, 3].

5.10.35 Utilizando los polinomios de Tchebyshev, construya un polinomio interpolador de Tchebyshev de orden nueve para la función $t^{8.37}\tan(0.1t)e^{-1.5t}$ en el intervalo [0, 14].

www.FreeLibros.me

www.FreeLibros.me

Capítulo 6

Derivación e integración numérica

6.1 Introducción

La solución apropiada de una ecuación diferencial es, por supuesto, su solución exacta; sin embargo, en general, las ecuaciones diferenciales de gran complejidad se tienen que resolver utilizando métodos numéricos aproximados. Entre los métodos numéricos más importantes se encuentran los conocidos como métodos de diferencias finitas.

Por supuesto, cualquier fórmula de interpolación numérica sirve para evaluar la derivada de una función, sin importar si la función viene en forma tabular o tiene una fórmula explícita. Ahora, si se supone que los puntos donde la función está definida se encuentran igualmente espaciados, esto permite obtener fórmulas sencillas para las aproximaciones de las derivadas.

Existen varias razones por las cuales puede ser necesario, o deseable, realizar el cálculo por aproximación numérica para evaluar una integral definida, en lugar de un análisis matemático. Por ejemplo, cuando es difícil o imposible encontrar una fórmula matemática para la integral, o si el problema se puede resolver analíticamente y la función es demasiado complicada para el cálculo eficiente por computadora. También cuando se necesite que un programa de integración se pueda usar para una función general sin análisis matemático especial en cada ocasión.

6.2 Derivación numérica

En esta sección se trabajará en la aproximación de la derivada de una función en un punto dado, conociendo el comportamiento de la función en algunos puntos vecinos. La forma más sencilla es aproximar la función por derivar mediante un polinomio interpolador y derivar este polinomio [Burden *et al.*, 2002]. Aunque la derivada del polinomio puede ser muy diferente a la función por derivar, como el polinomio tiende a tener muchas oscilaciones (cuando el número de puntos donde se interpola es grande), éstas se reducen si se toma en cuenta un número pequeño de puntos para construir el polinomio interpolador.

www.FreeLibros.me

EJEMPLO 6.1

El ejemplo 5.8 ilustra el caso donde se interpola la función $f(x)=\exp(-x^2)$ en el intervalo $[-5, +5]$ con 9 puntos de interpolación, usando puntos igualmente espaciados, con $n=8$. El polinomio interpolador tiene la siguiente forma:

$$p(x)=0.0001940349009x^8-0.009245209632x^6+0.1361270733x^4-0.6967161360x^2+1$$

La figura 6.1 presenta la gráfica de la derivada de $f(x)$ y de la derivada de polinomio interpolador $p(x)$. Ambas derivadas se dan a continuación:

$$f'(x)=-2x\exp(-x^2)$$

$$p'(x)=0.001552279207x^7-0.05547125779x^5+0.5445082932x^3-1.393432272x$$

La línea continua representa la gráfica de la derivada de la función analítica y la línea punteada representa la derivada del polinomio. Se puede observar que, en un intervalo alrededor del origen, las gráficas son similares, pero conforme el intervalo se aleja, las gráficas difieren bastante.

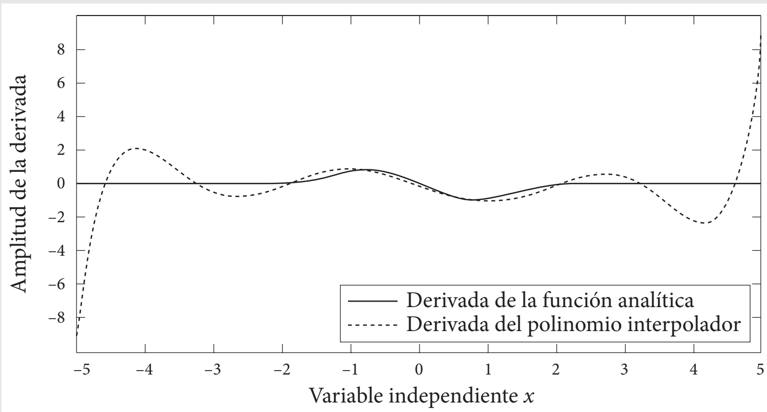


Figura 6.1 Gráfica de la derivada de la función evaluada y del polinomio interpolador.

No todo es tan malo con esta aproximación. Como se verá más adelante, es posible aproximar la derivada de la función mediante la derivada del polinomio interpolador; pero una buena idea es aproximar la derivada sólo en los puntos de interpolación. Para iniciar, se tiene que la definición de la derivada de una función f en $x=x_0$ está dada por

$$f'(x_0)=\lim_{x\to x_0}\frac{f(x)-f(x_0)}{x-x_0}=\lim_{h\to 0}\frac{f(x_0+h)-f(x_0)}{h} \quad \textbf{(6.1)}$$

Si h es suficientemente pequeño, se tiene que una buena aproximación a la derivada de f en $x=x_0$ está dada por

$$\frac{f(x_0+h)-f(x_0)}{h}$$

Esta aproximación se conoce como *aproximación de la derivada por la derecha*. Geométricamente, se tiene que $\frac{f(x_0+h)-f(x_0)}{h}$ es la pendiente de la recta secante que pasa por los puntos $(x_0, f(x_0))$ y

www.FreeLibros.me

$(x_0+h, f(x_0+h))$ mientras que $f'(x_0)$ es la pendiente de la recta tangente a f en x_0. De manera similar, se puede construir la *aproximación de la derivada por la izquierda*. Ésta es

$$\frac{f(x_0)-f(x_0-h)}{h}$$

Para determinar el error cometido al aproximar la derivada, se considera el polinomio interpolador. Se tiene que si $f \in C^{(n+1)}[a, b]$. Entonces

$$f(x)=p(x)+(x-x_0)(x-x_1)\cdots(x-x_n)\frac{f^{(n+1)}(\xi)}{(n+1)!}$$

Dado que se va a derivar esta expresión, se necesita además que $f \in C^{(n+2)}[a, b]$. Se tiene entonces que

$$f'(x)=p'(x)+\frac{d}{dx}\left[(x-x_0)(x-x_1)\cdots(x-x_n)\frac{f^{(n+1)}(\xi)}{(n+1)!}\right] \tag{6.2}$$

El valor más pequeño de n que se puede considerar en (6.2) es $n=1$, ya que para $n=0$ se tiene que el polinomio es constante y, por tanto, la derivada sería cero.

Considerando esta última fórmula con $n=1$, se obtiene

$$\begin{aligned} f'(x)&=p'(x)+\frac{d}{dx}\left[(x-x_0)(x-x_1)\frac{f''(\xi)}{2}\right] \\ &=p'(x)+\frac{1}{2}(x-x_0)f''(\xi)+\frac{1}{2}(x-x_1)f''(\xi)+\frac{1}{2}(x-x_0)(x-x_1)f'''(\xi)\frac{d\xi}{dx} \end{aligned}$$

ya que $\xi=\xi(x)$ depende de la elección de x. Dado que no se conoce ξ como función de x, la derivada $\frac{d\xi}{dx}$ también es un elemento desconocido en esta expresión. Para evitar este problema, si se evalúa la derivada en $x=x_0$ o en $x=x_1$, el último término se anula. Entonces

$$\begin{aligned} f'(x_0)&=p'(x_0)+\frac{1}{2}(x_0-x_0)f''(\xi)+\frac{1}{2}(x_0-x_1)f''(\xi)+\frac{1}{2}(x_0-x_0)(x_0-x_1)f'''(\xi)\frac{d\xi}{dx} \\ &=p'(x_0)+\frac{1}{2}(x_0-x_1)f''(\xi) \end{aligned} \tag{6.3}$$

y

$$\begin{aligned} f'(x_1)&=p'(x_1)+\frac{1}{2}(x_1-x_0)f''(\xi)+\frac{1}{2}(x_1-x_1)f''(\xi)+\frac{1}{2}(x_1-x_0)(x_1-x_1)f'''(\xi)\frac{d\xi}{dx} \\ &=p'(x_1)+\frac{1}{2}(x_1-x_0)f''(\xi) \end{aligned} \tag{6.4}$$

Considerando de aquí en adelante que el espacio entre los puntos es h, entonces (6.3) y (6.4) se reducen a

$$f'(x_0)=p'(x_0)-\frac{1}{2}hf''(\xi)$$

$$f'(x_1)=p'(x_1)+\frac{1}{2}hf''(\xi)$$

Usando la formulación de *diferencias divididas de Newton* dada por la ecuación (5.11) para evaluar la derivada de $p(x)$, se tiene

www.FreeLibros.me

$$
\begin{aligned}
p(x) &= \sum_{k=0}^{1} f[x_0, x_1, \ldots, x_k](x-x_0)(x-x_1)\cdots(x-x_{k-1}) \\
&= f[x_0] + f[x_0, x_1](x-x_0) \\
&= f_0 + \left(\frac{f_1 - f_0}{h}\right)(x-x_0)
\end{aligned}
$$

y, por tanto, la expresión para la derivada es

$$
p'(x) = \frac{d\left(f_0 + \left(\frac{f_1 - f_0}{h}\right)(x-x_0)\right)}{dx} = \frac{f_1 - f_0}{h}
$$

por lo que

$$
p'(x_0) = \frac{f_1 - f_0}{h} \ \text{ y } \ p'(x_1) = \frac{f_1 - f_0}{h}
$$

Así (6.3) y (6.4), se reducen a

$$
f'(x_0) = \frac{f_1 - f_0}{h} - \frac{1}{2} h f''(\xi) \tag{6.5}
$$

$$
f'(x_1) = \frac{f_1 - f_0}{h} + \frac{1}{2} h f''(\xi) \tag{6.6}
$$

La fórmula (6.5) se conoce como *aproximación a la derivada por la derecha*, mientras que (6.6) es la *aproximación por la izquierda*. A estas fórmulas se les llama *aproximaciones de dos puntos*. Una ventaja de esta forma de desarrollar las aproximaciones es que se tienen fórmulas para el error de aproximación. Si la segunda derivada está acotada por $K_2 \geq 0$, se tiene que los errores de aproximación satisfacen

$$
\left| f'(x_0) - \frac{f_1 - f_0}{h} \right| \leq \frac{1}{2} h |f''(\xi)| \leq \frac{K_2}{2} h \ \text{ y } \ \left| f'(x_1) - \frac{f_1 - f_0}{h} \right| \leq \frac{1}{2} h |f''(\xi)| \leq \frac{K_2}{2} h
$$

Dado que el exponente de h es 1 (6.5) y (6.6), se llaman *aproximaciones de primer orden* o de *orden h*. También se conocen como *aproximaciones de dos puntos*. Usando la fórmula para la derivada (6.2) con $n = 2$, se tiene

$$
\begin{aligned}
f'(x) &= p'(x) + \frac{d}{dx}\left[(x-x_0)(x-x_1)(x-x_2)\frac{f'''(\xi)}{6}\right] \\
&= p'(x) + \frac{1}{6}(x-x_1)(x-x_2) f'''(\xi) + \frac{1}{6}(x-x_0)(x-x_2) f'''(\xi) \\
&\quad + \frac{1}{6}(x-x_0)(x-x_1) f'''(\xi) + \frac{1}{6}(x-x_0)(x-x_1)(x-x_2) f^{(iv)}(\xi)\frac{d\xi}{dx}
\end{aligned}
$$

De nuevo se tienen tres opciones para evaluar la derivada de la función. Éstas son x_0, x_1 y x_2. Se tiene entonces

$$
f'(x_0) = p'(x_0) + \frac{1}{6}(x_0 - x_1)(x_0 - x_2) f'''(\xi)
$$

$$
f'(x_1) = p'(x_1) + \frac{1}{6}(x_1 - x_0)(x_1 - x_2) f'''(\xi)
$$

$$
f'(x_2) = p'(x_2) + \frac{1}{6}(x_2 - x_0)(x_2 - x_1) f'''(\xi)
$$

www.FreeLibros.me

Considerando que los puntos están igualmente espaciados, se llega a

$$f'(x_0) = p'(x_0) + \frac{h^2}{3} f'''(\xi)$$
$$f'(x_1) = p'(x_1) - \frac{h^2}{6} f'''(\xi)$$
$$f'(x_2) = p'(x_2) + \frac{h^2}{3} f'''(\xi)$$

También se tiene que

$$\begin{aligned} p(x) &= f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\ &= f_0 + \frac{1}{h}(f_1 - f_0)(x - x_0) + \frac{1}{2h^2}(f_2 - 2f_1 + f_0)(x - x_0)(x - x_1) \end{aligned}$$

y, por tanto,

$$p'(x) = \frac{1}{h}(f_1 - f_0) + \frac{1}{2h^2}(f_2 - 2f_1 + f_0)[(x - x_0) + (x - x_1)]$$

por lo que

$$f'(x_0) = \frac{1}{h}(f_1 - f_0) + \frac{1}{2h^2}(f_2 - 2f_1 + f_0)(x_0 - x_1) + \frac{h^2}{3} f'''(\xi)$$
$$f'(x_1) = \frac{1}{h}(f_1 - f_0) + \frac{1}{2h^2}(f_2 - 2f_1 + f_0)(x_1 - x_0) - \frac{h^2}{6} f'''(\xi)$$
$$f'(x_2) = \frac{1}{h}(f_1 - f_0) + \frac{1}{2h^2}(f_2 - 2f_1 + f_0)[(x_2 - x_0) + (x_2 - x_1)] + \frac{h^2}{3} f'''(\xi)$$

Simplificando,

$$f'(x_0) = \frac{-f_2 + 4f_1 - 3f_0}{2h} + \frac{h^2}{3} f'''(\xi)$$
$$f'(x_1) = \frac{f_2 - f_0}{2h} - \frac{h^2}{6} f'''(\xi)$$
$$f'(x_2) = \frac{f_0 - 4f_1 + 3f_2}{2h} + \frac{h^2}{3} f'''(\xi)$$

Las aproximaciones se llaman *aproximaciones de tres puntos por la derecha*, *central* e *izquierda,* respectivamente. Además, se dice que éstas son *aproximaciones de segundo orden* o de *orden* h^2. Usando la fórmula para la derivada para la aproximación con $n = 3$, se tiene

$$f'(x) = p'(x) + \frac{d}{dx}\left[(x - x_0)(x - x_1)(x - x_2)(x - x_3)\frac{f^{(iv)}(\xi)}{24}\right]$$

Es decir,

$$\begin{aligned} f'(x) = p'(x) &+ \frac{1}{24}(x - x_1)(x - x_2)(x - x_3) f^{(iv)}(\xi) + \frac{1}{24}(x - x_0)(x - x_2)(x - x_3) f^{(iv)}(\xi) \\ &+ \frac{1}{24}(x - x_0)(x - x_1)(x - x_3) f^{(iv)}(\xi) + \frac{1}{24}(x - x_0)(x - x_1)(x - x_2) f^{(iv)}(\xi) \\ &+ \frac{1}{24}(x - x_0)(x - x_1)(x - x_2)(x - x_3) f^{(v)}(\xi)\frac{d\xi}{dx} \end{aligned}$$

www.FreeLibros.me

Ahora se tienen cuatro opciones para evaluar la derivada de la función. Éstas son x_0, x_1, x_2 y x_3. Se tiene entonces que

$$f'(x_0)=p'(x_0)+\frac{1}{24}(x_0-x_1)(x_0-x_2)(x_0-x_3)f^{(iv)}(\xi)$$

$$f'(x_1)=p'(x_1)+\frac{1}{24}(x_1-x_0)(x_1-x_2)(x_1-x_3)f^{(iv)}(\xi)$$

$$f'(x_2)=p'(x_2)+\frac{1}{24}(x_2-x_0)(x_2-x_1)(x_2-x_3)f^{(iv)}(\xi)$$

$$f'(x_3)=p'(x_3)+\frac{1}{24}(x_3-x_0)(x_3-x_1)(x_3-x_2)f^{(iv)}(\xi)$$

Considerando que los puntos están igualmente espaciados, se tiene

$$f'(x_0)=\frac{-11f_0+18f_1-9f_2+2f_3}{6h}-\frac{1}{4}h^3f^{(iv)}(\xi)$$

$$f'(x_1)=\frac{-5f_0+6f_1-3f_2+2f_3}{6h}+\frac{1}{12}h^3f^{(iv)}(\xi)$$

$$f'(x_2)=\frac{f_0-6f_1+3f_2+2f_3}{6h}-\frac{1}{12}h^3f^{(iv)}(\xi)$$

$$f'(x_3)=\frac{-2f_0+9f_1-18f_2+11f_3}{6h}+\frac{1}{4}h^3f^{(iv)}(\xi)$$

Las aproximaciones se llaman aproximaciones de cuatro puntos o de tercer orden. En general, se puede usar el polinomio interpolador con $n+1$ puntos, y se obtendrán $n+1$ aproximaciones de la primera derivada. Éstas se dice que son *aproximaciones de orden* $O(h^n)$ o simplemente de *orden n*. La tabla 6.1 presenta las aproximaciones de la derivada con $n=1, 2, 3$.

Tabla 6.1 Aproximaciones de la derivada para n = 1, 2, 3.

n	$x=x_0$	$x=x_1$	$x=x_2$	$x=x_3$	Orden
1	$\frac{f_1-f_0}{h}$	$\frac{f_1-f_0}{h}$			$O(h)$
2	$\frac{-f_2+4f_1-3f_0}{2h}$	$\frac{f_2-f_0}{2h}$	$\frac{f_0-4f_1+3f_2}{2h}$		$O(h^2)$
3	$\frac{-11f_0+18f_1-9f_2+2f_3}{6h}$	$\frac{-5f_0+6f_1-3f_2+2f_3}{6h}$	$\frac{f_0-6f_1+3f_2+2f_3}{6h}$	$\frac{-2f_0+9f_1-18f_2+11f_3}{6h}$	$O(h^3)$

Para la aproximación de las derivadas de orden n, es necesario usar polinomios interpoladores con al menos $n+1$ puntos. Si se usan exactamente $n+1$ puntos, se tiene que

$$f^{(n)}(x)=p^{(n)}(x)+\frac{d^n\left[(x-x_0)(x-x_1)\cdots(x-x_n)\frac{f^{n+1}(\xi)}{(n+1)!}\right]}{dx^n}$$

Pero

$$\frac{d^n\left[(x-x_0)(x-x_1)\cdots(x-x_n)\frac{f^{n+1}(\xi)}{(n+1)!}\right]}{dx^n}=n![(x-x_0)+(x-x_1)+\cdots+(x-x_n)]\frac{f^{n+1}(\xi)}{(n+1)!}$$

www.FreeLibros.me

Simplificando, se obtiene

$$\frac{d^n\left[(x-x_0)(x-x_1)\cdots(x-x_n)\dfrac{f^{n+1}(\xi)}{(n+1)!}\right]}{dx^n}=[(x-x_0)+(x-x_1)+\cdots+(x-x_n)]\frac{f^{n+1}(\xi)}{(n+1)}$$

Por tanto,

$$p^{(n)}(x)=\frac{d^n\left[\sum_{k=0}^{n}f[x_0,\, x_1,\ldots,\, x_k](x-x_0)(x-x_1)\cdots(x-x_{k-1})\right]}{dx^n}$$

Con lo que se obtiene que, por ejemplo

$$p^{(n)}(x)=n!\,f[x_0,\, x_1,\ldots,\, x_n]=n!\frac{1}{n!h^n}\Delta^n y_0=\frac{1}{h^n}\Delta^n y_0$$

donde $\Delta^n y_0 = y_n+\alpha y_{n-1}+\cdots+\omega y_0$ por ejemplo $\Delta^2 y_0 = y_2 - 2y_1 + y_0$, por lo que se tiene

$$f^{(n)}(x)=\frac{1}{h^n}\Delta^n y_0+[(x-x_0)+(x-x_1)+\cdots+(x-x_n)]\frac{f^{n+1}(\xi)}{(n+1)!}$$

$$f^{(n)}(x_i)=\frac{1}{h^n}\Delta^n y_0+\frac{(2i-n)(n+1)h}{2}\frac{f^{n+1}(\xi)}{(n+1)!}$$

$$f^{(n)}(x_i)=\frac{1}{h^n}\Delta^n y_0+\frac{(2i-n)}{2n!}hf^{n+1}(\xi)$$

De esta expresión se sigue que

$$f^{(n)}(x_i)=\frac{1}{h^n}\Delta^n y_0+O(h)$$

Estas fórmulas se pueden generalizar fácilmente a derivadas parciales de funciones en varias variables. Si, por ejemplo, $u=u(x,y)$, algunas aproximaciones de $\frac{\partial u}{\partial x}(x,y)$ son

$$\frac{u(x+h,y)-u(x,y)}{h}$$

$$\frac{u(x,y)-u(x-h,y)}{h}$$

$$\frac{u(x+h,y)-u(x-h,y)}{2h}$$

Similarmente, para $\frac{\partial^2 u}{\partial x\partial y}(x,\, y)$, se muestran algunas aproximaciones usando derivadas por la derecha para x y derivadas por la derecha, central y por la izquierda para y.

$$\frac{u(x+h,y+k)-u(x,y+k)-u(x+h,y)+u(x,y)}{hk}$$

$$\frac{u(x+h,y)-u(x,y)-u(x+h,y-k)+u(x,y-k)}{hk}$$

$$\frac{u(x+h,y+k)-u(x,y+k)-u(x+h,y-k)+u(x,y-k)}{2hk}$$

www.FreeLibros.me

EJEMPLO 6.2

Un automóvil recorre una pista en 65 segundos. La distancia recorrida por el automóvil se determina cada 5 segundos. Los datos se presentan en la siguiente tabla

Tabla 6.2 Tabla de datos.

Tiempo	0	5	10	15	20	25	30	35	40	45	50	55	60	65
Distancia	0	54	115	175	250	330	400	460	516	566	628	698	774	844

Se determinarán la velocidad y la aceleración cada 5 segundos. Se tiene que la velocidad y la aceleración están dadas por la primera y segunda derivada de la posición, respectivamente. Usando derivadas por la derecha para determinar la velocidad, se obtienen los resultados dados en la tabla 6.3. Cabe hacer notar que no es posible determinar la velocidad para $t = 65$.

Tabla 6.3 Tabla de tiempos y velocidades calculadas con la primera derivada por la derecha.

Tiempo	0	5	10	15	20	25	30	35	40	45	50	55	60	65
Velocidad	18.8	12.2	12	15	16	14	12	11.2	10	12.4	14	15.2	14	

Usando derivadas centrales de segundo orden se tiene que para $t = 0$ y $t = 65$, no es posible calcular la derivada, y se obtienen los resultados dados en la tabla 6.4.

Tabla 6.4 Tabla de tiempos y velocidades calculadas con derivadas centrales.

Tiempo	0	5	10	15	20	25	30	35	40	45	50	55	60	65
Velocidad		11.5	12.1	13.5	15.5	15	13	11.6	10.6	11.2	13.2	14.6	14.6	

Se pueden usar derivadas de mayor orden, pero, en general, la cantidad de trabajo efectuada para calcularlas es mayor que la precisión obtenida. Calculando la aceleración por medio de segundas derivadas, se obtiene

Tabla 6.5 Tabla de tiempos y aceleración calculadas con derivadas de segundo orden por la derecha.

Tiempo	0	5	10	15	20	25	30	35	40	45	50	55	60	65
Aceleración		–1.88	–0.32	0.64	–0.4	–0.6	0	0.24	–0.8	0.72	–0.16	–0.08	–0.48	

6.3 Integración numérica

En esta sección se presentan los métodos numéricos para evaluar una integral definida, con la siguiente estructura:

$$I = \int_a^b f(x)dx$$

Los métodos considerados aquí se desarrollan tomando una función simple $Q_n(x)$ que tiene el mismo valor que $f(x)$ en un número de puntos elegidos, $x_i\,(i = 0, 1, \ldots, n)$ y usando la integral de $Q_n(x)$ como una aproximación de la integral de $f(x)$. Las funciones $Q_n(x)$ deben ser, por consiguiente, fáciles de integrar. Los puntos x_i, en los que se tendrá que $Q_n(x_i) = f(x_i)$, se conocen como *nodos de la fórmula de integración*. A los valores de $f(x_i)$ se les da la notación f_i. Si los nodos elegidos son equidistantes, entonces se puede obtener una serie de fórmulas conocidas como *fórmulas de Newton-Cotes*. Las dos formulaciones más sencillas de esta clase son la *regla trapezoidal* y la *regla de Simpson*. Sin embargo, estas

www.FreeLibros.me

formulaciones pierden precisión debido a la restricción a puntos equidistantes. Se pueden obtener formulaciones dos veces más precisas que las formulaciones de Newton-Cotes si los nodos son especialmente elegidos para dar la máxima precisión posible. Estas formulaciones se conocen como *formulación de cuadraturas gaussianas*. Los nodos de estas formulaciones son los ceros de ciertos polinomios ortogonales. Se debe notar que la formulación es sólo más precisa en términos de la definición matemática particular de precisión. Puede suceder fácilmente que una fórmula de Newton-Cotes dé un valor numérico más cercano que la fórmula teórica gaussiana más precisa.

Otro método adicional es el que se basa en el uso de la regla trapezoidal con muchos tamaños de intervalo diferentes. Entonces se usa la técnica de extrapolación de Richardson para reducir el error consecutivamente. Este método, conocido como *integración de Romberg*, es muy apropiado para usarse en programas de cómputo, y se puede demostrar que converge para cualquier función continua $f(x)$. Antes de proporcionar ejemplos de cómo usar las fórmulas de diferencias finitas para la derivación e integración, es mejor observar gráficamente el problema para tener una mejor apreciación de las dificultades que presenta. Considere los dos ejemplos de la figura 6.2, *a* y *b*. La diferencia entre estas dos figuras es que el segundo punto tiene un valor de la función ligeramente diferente.

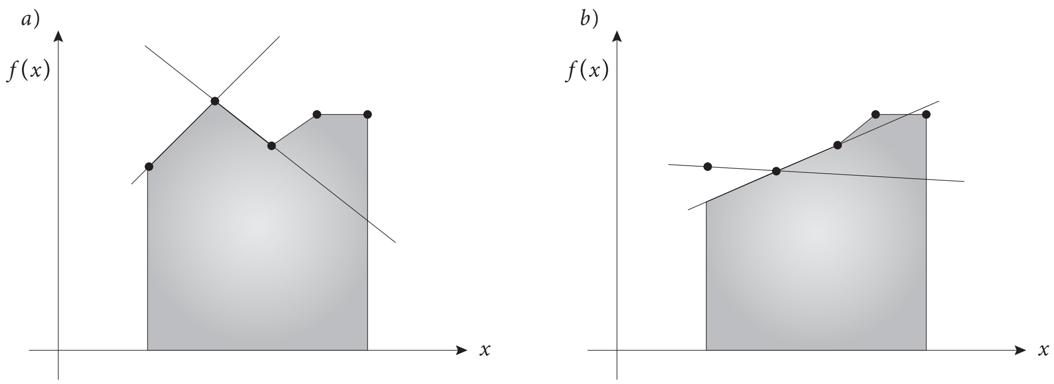


Figura 6.2 *a*) Error en la integral, *b*) Error en la derivada.

En este caso se ha exagerado la diferencia de manera que se aprecie fácilmente en el diagrama; pero los valores normalmente están sujetos a algún tipo de error de redondeo o experimental. El efecto de este error sobre la aproximación de la derivada por medio de dos puntos adyacentes es notable, como lo demuestra el diagrama. Sin embargo, si la integral se aproxima mediante un área cerrada, se puede ver que el error en la integral es relativamente pequeño.

Estas observaciones pueden ser fundamentales en el análisis matemático de los métodos de diferencias finitas. Esto demuestra que la diferenciación por medio de la fórmula de diferencias finitas puede llevar a grandes errores, aunque el proceso de integración esté bien condicionado. Por tanto, hasta donde sea posible, los problemas serán formulados de manera que no sea necesaria la diferenciación de las fórmulas de diferencias finitas, o aproximación a las funciones.

6.4 Fórmulas de Newton-Cotes

Al igual que para la aproximación de las derivadas de una función, se usará el polinomio interpolador para aproximar la integral definida de una función. La idea básica es que, dada una función a integrar sobre un intervalo, se define el polinomio interpolador en ese intervalo y se integra el polinomio interpolador. Se espera que ésta sea una buena aproximación a la integral. Se tienen varias formulaciones para el polinomio interpolador. Las más sencillas son las que se obtienen con las diferencias de Newton. Con esta aproximación se obtendrán las fórmulas de Newton-Cotes, que pueden ser cerradas o abiertas, dependiendo de si el polinomio interpolador construido incluye o no los extremos del intervalo de integración.

www.FreeLibros.me

6.4.1 Fórmulas cerradas de Newton-Cotes

Para iniciar, se considera el problema de aproximar a la integral [Nakamura, 1992], [Maron *et al.*, 1995], [Burden *et al.*, 2002],

$$\int_a^b f(x)dx$$

definiendo $x_0 = a,\ x_1 = b,\ h = b - a$ y considerando el polinomio interpolador de grado cero basado en el punto x_0. El polinomio es

$$p(x) = f(x_0)$$

Se infiere además, de (6.5), que,

$$f(x) = f(x_0) + (x - x_0)\frac{f'(\xi)}{2}$$

Integrando a ambos lados desde $x_0 = a$ hasta $x_1 = b$, se sigue que

$$\begin{aligned}\int_a^b f(x)dx &= \int_{x_0}^{x_1}\left(f(x_0) + (x - x_0)\frac{f'(\xi)}{2}\right)dx \\ &= f(x_0)\int_{x_0}^{x_1} dx + \int_{x_0}^{x_1}(x - x_0)\frac{f'(\xi)}{2}dx \\ &= (x_1 - x_0)f(x_0) + \int_{x_0}^{x_1}(x - x_0)\frac{f'(\xi)}{2}dx\end{aligned}$$

Usando el teorema del valor medio para integrales y el hecho de que $h = x_1 - x_0$ se tiene

$$\int_a^b f(x)dx = hf(x_0) + \frac{f'(\xi)}{2}\int_{x_0}^{x_1}(x - x_0)dx$$

Así, finalmente

$$\int_a^b f(x)dx = hf(x_0) + \frac{h^2 f'(\xi)}{2} \tag{6.7}$$

De forma similar, se podría construir el polinomio de grado cero basado en el punto $x_1 = b$ y obtener

$$\int_a^b f(x)dx = hf(x_1) - \frac{h^2 f'(\xi)}{2} \tag{6.8}$$

Las fórmulas (6.7) y (6.8) se conocen como la *regla rectangular por la izquierda y por la derecha*, respectivamente. La figura (6.3) muestra la aproximación mediante la regla rectangular por la izquierda. La sección 6.8.1 proporciona el código desarrollado en Matlab para la regla rectangular por la izquierda y la sección 6.8.2 para la regla rectangular por la derecha.

Generalizando, si se desea aproximar

$$\int_a^b f(x)dx$$

donde $f \in C^{(n+1)}[a, b]$, se establece el número n de subintervalos de $[a, b]$. A continuación se define $h = \dfrac{b - a}{n}$ y $x_i = a + ih,\ i = 0, 1, \ldots, n$ y se construye el polinomio interpolador que pasa por los puntos $(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n))$. La formulación con diferencias adelantadas es

$$f(x) = \sum_{k=0}^{n}\Delta^k f_0\binom{s}{k} + h^{n+1}\binom{s}{n+1}f^{(n+1)}(\xi), \quad x = x_0 + sh \tag{6.9}$$

www.FreeLibros.me

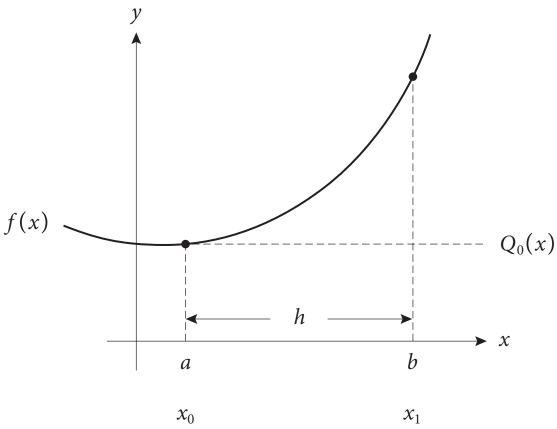


Figura 6.3 Gráfica de la regla rectangular.

Integrando ambos lados desde $a = x_0$ hasta $b = x_n$, y tomando en cuenta que la variable en el lado derecho de (6.9) es s, y que se debe integrar con respecto a x, haciendo el cambio de variable $x = x_0 + sh$ en la integral se obtiene

$$\begin{aligned}\int_{x_0}^{x_n} f(x)dx &= \int_{x_0}^{x_n}\left(\sum_{k=0}^{n}\Delta^k f_0\binom{s}{k} + h^{n+1}\binom{s}{n+1}f^{(n+1)}(\xi)\right)dx \\ &= \int_0^n\left(\sum_{k=0}^{n}\Delta^k f_0\binom{s}{k} + h^{n+1}\binom{s}{n+1}f^{(n+1)}(\xi)\right)hds\end{aligned}$$

Intercambiando el orden entre la integración y la sumatoria,

$$\int_{x_0}^{x_n} f(x)dx = h\sum_{k=0}^{n}\Delta^k f_0\int_0^n\binom{s}{k}ds + h^{n+2}\int_0^n\binom{s}{n+1}f^{(n+1)}(\xi)ds$$

Definiendo además

$$b_{nk} = \int_0^n\binom{s}{k}ds = \int_0^n\frac{s(s-1)(s-2)\cdots(s-k+1)}{k!}ds,$$

entonces

$$\int_{x_0}^{x_n} f(x)dx = h\sum_{k=0}^{n}\Delta^k f_0 b_{nk} + h^{n+2}\int_0^n\binom{s}{n+1}f^{(n+1)}(\xi)ds \tag{6.10}$$

6.4.1.1 *Considerando diferentes valores de* n

Tomando $n = 1$ se tiene

$$\begin{aligned}\int_{x_0}^{x_1} f(x)dx &= h\sum_{k=0}^{1}\Delta^k f_0 b_{1k} + h^3\int_0^1\binom{s}{2}f''(\xi)ds \\ &= h\left(\Delta^0 f_0 b_{10} + \Delta^1 f_0 b_{11}\right) + h^3\int_0^1\binom{s}{2}f''(\xi)ds\end{aligned}$$

ahora

$$b_{10} = \int_0^1\binom{s}{0}ds = \int_0^1 ds = 1 \text{ y } b_{11} = \int_0^1\binom{s}{1}ds = \int_0^1 sds = \frac{1}{2}$$

www.FreeLibros.me

por lo que

$$\int_{x_0}^{x_1} f(x)dx = h\left(\Delta^0 f_0 + \frac{1}{2}\Delta^1 f_0\right) + h^3\int_0^1 \binom{s}{2} f''(\xi)ds$$

$$= h\left(f_0 + \frac{1}{2}(f_1 - f_0)\right) + h^3\int_0^1 \binom{s}{2} f''(\xi)ds$$

$$= h\left(f_0 + \frac{1}{2}(f_1 - f_0)\right) + h^3\int_0^1 \binom{s}{2} f''(\xi)ds$$

$$= \frac{h}{2}(f_0 + f_1) + h^3\int_0^1 \binom{s}{2} f''(\xi)ds$$

La integral

$$\int_0^1 \binom{s}{2} f''(\xi)ds$$

debe ser evaluada usando el teorema del valor medio para integrales. Si se tiene que

$$\int_0^1 \binom{s}{2} f''(\xi)ds = f''(\xi)\int_0^1 \binom{s}{2} ds$$

$$= f''(\xi)\int_0^1 \frac{s(s-1)}{2} ds$$

$$= -\frac{1}{12} f''(\xi)$$

entonces

$$\int_{x_0}^{x_1} f(x)dx = \frac{h}{2}(f_0 + f_1) - \frac{h^3}{12} f''(\xi)$$

Este método se conoce como la regla trapezoidal. La figura 6.4 ilustra este resultado y la sección 6.8.3 proporciona el código desarrollado en Matlab para esta regla de integración.

Con $n = 2$ se tiene

$$\int_{x_0}^{x_2} f(x)dx = h\sum_{k=0}^{2} \Delta^k f_0 b_{2k} + h^4\int_0^2 \binom{s}{3} f'''(\xi)ds$$

$$= h\left(\Delta^0 f_0 b_{20} + \Delta^1 f_0 b_{21} + \Delta^2 f_0 b_{22}\right) + h^4\int_0^2 \binom{s}{3} f'''(\xi)ds$$

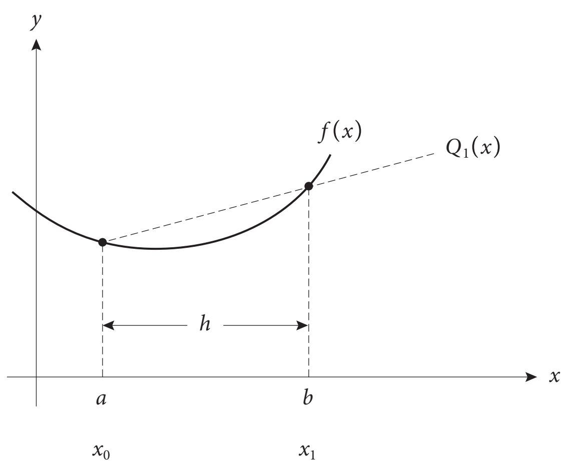


Figura 6.4 Gráfica de la regla trapezoidal.

www.FreeLibros.me

Ahora

$$b_{20}=\int_0^2\binom{s}{0}ds=\int_0^2 ds=2,\; b_{21}=\int_0^2\binom{s}{1}ds=\int_0^2 s\,ds=2 \text{ y } b_{22}=\int_0^2\binom{s}{2}ds=\int_0^2\frac{s(s-1)}{2}ds=\frac{1}{3}$$

por lo que

$$\begin{aligned}\int_{x_0}^{x_2}f(x)dx&=h\left(2\Delta^0 f_0+2\Delta^1 f_0+\frac{1}{3}\Delta^2 f_0\right)+h^4\int_0^2\binom{s}{3}f'''(\xi)ds\\&=h\left(2f_0+2(f_1-f_0)+\frac{1}{3}(f_2-2f_1+f_0)\right)+h^4\int_0^2\binom{s}{3}f'''(\xi)ds\\&=\frac{h}{3}(f_0+4f_1+f_2)+h^4\int_0^2\binom{s}{3}f'''(\xi)ds\end{aligned}$$

Se puede demostrar que

$$\int_{x_0}^{x_2}f(x)dx=\frac{h}{3}(f_0+4f_1+f_2)-\frac{h^5}{90}f^{(iv)}(\xi)$$

Este método se conoce como *Regla de Simpson* 1/3, y se ilustra en la figura 6.5; la sección 6.8.4 proporciona el código desarrollado en Matlab de esta regla de integración.

Con $n=3$ se tiene

$$\begin{aligned}\int_{x_0}^{x_3}f(x)dx&=h\sum_{k=0}^{3}\Delta^k f_0 b_{3k}+h^5\int_0^3\binom{s}{4}f^{(iv)}(\xi)ds\\&=h\left(\Delta^0 f_0 b_{30}+\Delta^1 f_0 b_{31}+\Delta^2 f_0 b_{32}+\Delta^3 f_0 b_{33}\right)+h^5\int_0^3\binom{s}{4}f^{(iv)}(\xi)ds\end{aligned}$$

Ahora

$$b_{30}=\int_0^3\binom{s}{0}ds=\int_0^3 ds=3$$

$$b_{31}=\int_0^3\binom{s}{1}ds=\int_0^3 s\,ds=\frac{9}{2}$$

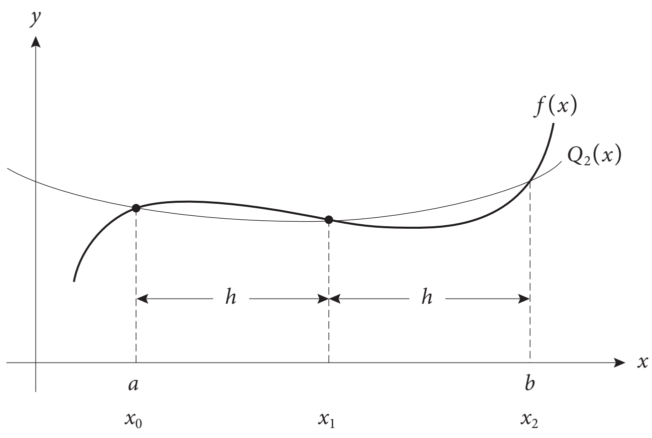


Figura 6.5 Gráfica de la regla de Simpson 1/3.

www.FreeLibros.me

$$b_{32} = \int_0^3 \binom{s}{2} ds = \int_0^3 \frac{s(s-1)}{2} ds = \frac{9}{4}$$

$$b_{33} = \int_0^3 \binom{s}{3} ds = \int_0^3 \frac{s(s-1)(s-2)}{3!} ds = \frac{3}{8}$$

por lo que

$$\int_{x_0}^{x_3} f(x)dx = h\left(3\Delta^0 f_0 + \frac{9}{2}\Delta^1 f_0 + \frac{9}{4}\Delta^2 f_0 + \frac{3}{8}\Delta^3 f_0\right) + h^5 \int_0^3 \binom{s}{4} f^{(iv)}(\xi)ds$$

$$= h\left(3f_0 + \frac{9}{2}(f_1 - f_0) + \frac{9}{4}(f_2 - 2f_1 + f_0) + \frac{3}{8}(f_3 - 3f_2 + 3f_1 - f_0)\right)$$

$$+ h^5 \int_0^3 \binom{s}{4} f^{(iv)}(\xi)ds$$

$$= \frac{3h}{8}(f_0 + 3f_1 + 3f_2 + f_3) + h^5 \int_0^3 \binom{s}{4} f^{(iv)}(\xi)ds$$

Evaluando la integral del lado derecho, se puede demostrar que

$$\int_{x_0}^{x_3} f(x)dx = \frac{3h}{8}(f_0 + 3f_1 + 3f_2 + f_3) - \frac{3h^5}{80} f^{(iv)}(\xi)$$

Este método se conoce como la *regla de Simpson de los tres octavos.* El código desarrollado en Matlab para esta regla de integración se proporciona en la sección 6.8.5. El tipo de reglas así obtenidas se conoce como las *fórmulas cerradas de Newton-Cotes*. Se tiene el siguiente teorema que nos permite estimar el error. Éste es

Teorema 6.1 Para n dado si n es par y $f \in C^{n+2}[x_0, x_n]$, y si n es impar y $f \in C^{n+1}[x_0, x_n]$, entonces

$$\int_{x_0}^{x_n} f(x)dx = h\sum_{k=0}^{n} b_{nk}\Delta^k f_0 + \frac{h^{n+2} f^{(n+1)}(\xi)}{(n+1)!}\int_0^n s(s-1)\cdots(s-n)ds, \quad n \text{ impar}$$

$$\int_{x_0}^{x_n} f(x)dx = h\sum_{k=0}^{n} b_{nk}\Delta^k f_0 + \frac{h^{n+3} f^{(n+2)}(\xi)}{(n+2)!}\int_0^n s^2(s-1)\cdots(s-n)ds, \quad n \text{ par}$$

donde

$$b_{nk} = \int_0^n \binom{s}{k} ds$$

$$\xi \in (x_0, x_n)$$

Observe que cuando n es par, el orden es $O(h^{n+3})$, pero cuando n es impar el orden es, simplemente $O(h^{n+2})$, por lo que es aconsejable usar sólo valores pares de n. Existen formas alternas para generar estas reglas. Como ejemplo, considere la siguiente formulación

$$\int_0^h f(x)dx = \sum_{i=0}^{1} a_i f_i + E, \; x_i = 0, h$$

o bien, en forma expandida se tiene

$$\int_0^h f(x)dx = a_0 f_0 + a_1 f_1 + E, \; (x := \quad x_0 = 0, \; x_1 = h)$$

www.FreeLibros.me

Si es necesario que esta aproximación sea exacta para polinomios de grado cero y uno, se debe cumplir que, para $f(x)=1$ y $f(x)=x$, la integral sea exacta, por lo que

$$\int_0^h f(x)dx = a_0 f(x_0) + a_1 f(x_1)$$

$$\int_0^h (1)dx = a_0(1) + a_1(1)$$

$$h = a_0 + a_1$$

También, para $f(x)=x$, se tiene

$$\int_0^h f(x)dx = a_0 f(x_0) + a_1 f(x_1)$$

$$\int_0^h (x)dx = a_0(0) + a_1(h)$$

$$\frac{h^2}{2} = 0 + h \cdot a_1$$

Con esto se forma un sistema de dos ecuaciones con dos incógnitas. Su solución es

$$a_0 = \frac{h}{2},\ a_1 = \frac{h}{2}$$

Así se tiene que

$$\int_0^h f(x)dx = a_0 f_0 + a_1 f_1 + E$$

La fórmula anterior corresponde a la regla trapezoidal. Cabe hacer notar que este desarrollo no permite calcular el error en forma explícita, como es el caso con el desarrollo a partir del polinomio interpolador.

6.4.2 Fórmulas abiertas de Newton-Cotes

La atención se ha centrado en las fórmulas cerradas de Newton-Cotes ya que, en general, son computacionalmente más eficientes, sobre todo en las formas compuestas. Sin embargo, algunas veces es conveniente usar fórmulas abiertas, las cuales no usan los puntos extremos a y b [Nakamura, 1992], [Maron *et al.*, 1995], [Burden *et al.*, 2002]. Estas fórmulas encuentran aplicaciones en la integración de ecuaciones diferenciales como fórmulas del tipo predictor.

Para desarrollar este tipo de fórmulas, se considerará la integral, pero ahora cambiando la definición de h. Ésta se define ahora como $h=\dfrac{b-a}{n+2}$, y además $x_i = a + ih$, $i = 0, 1, \ldots, n+1$. En este caso se tiene $x_{-1} = a$ y $x_{n+1} = b$. Considerando el polinomio interpolador que pasa por los puntos $(x_0, f(x_0))$, $(x_1, f(x_1))$,…, $(x_n, f(x_n))$, se tiene que

$$f(x) = \sum_{k=0}^{n} \Delta^k f_0 \binom{s}{k} + h^{n+1} \binom{s}{n+1} f^{(n+1)}(\xi),\ x = x_0 + sh$$

Integrando ambos lados desde $a = x_{-1}$ hasta $b = x_{n+1}$, y tomando en cuenta que la variable del lado derecho es s y que se debe integrar con respecto a x, haciendo el cambio de variable $x = x_0 + sh$ en la integral se obtiene

www.FreeLibros.me

$$\int_{x_{-1}}^{x_{n+1}} f(x)dx = \int_{x_{-1}}^{x_{n+1}} \left(\sum_{k=0}^{n} \Delta^k f_0 \binom{s}{k} + h^{n+1} \binom{s}{n+1} f^{(n+1)}(\xi) \right) dx$$
$$= \int_{-1}^{n+1} \left(\sum_{k=0}^{n} \Delta^k f_0 \binom{s}{k} + h^{n+1} \binom{s}{n+1} f^{(n+1)}(\xi) \right) hds$$

Intercambiando el orden entre la integración y la sumatoria, esto conduce a

$$\int_{x_{-1}}^{x_{n+1}} f(x)dx = h\sum_{k=0}^{n} \Delta^k f_0 \int_{-1}^{n+1} \binom{s}{k} ds + h^{n+2} \int_{-1}^{n+1} \binom{s}{n+1} f^{(n+1)}(\xi) ds$$

Definiendo además

$$c_{nk} = \int_{-1}^{n+1} \binom{s}{k} ds,$$

entonces

$$\int_{x_{-1}}^{x_{n+1}} f(x)dx = h\sum_{k=0}^{n} \Delta^k f_0 c_{nk} + h^{n+2} \int_{-1}^{n+1} \binom{s}{n+1} f^{(n+1)}(\xi) ds \tag{6.11}$$

Tomando $n = 0$, se tiene que

$$c_{00} = \int_{-1}^{1} \binom{s}{0} ds = \int_{-1}^{1} ds = 2$$

e

$$\int_{x_{-1}}^{x_1} f(x)dx = h\sum_{k=0}^{0} \Delta^k f_0 c_{0k} + h^2 \int_{-1}^{1} \binom{s}{1} f'(\xi) ds$$
$$= 2hf_0 + \frac{h^3}{3} f''(\xi)$$

Esta formulación se conoce como la *regla del punto medio*. Similarmente, con $n = 1$, se tiene

$$c_{10} = \int_{-1}^{2} \binom{s}{0} ds = \int_{-1}^{2} ds = 3$$
$$c_{11} = \int_{-1}^{2} \binom{s}{1} ds = \int_{-1}^{2} sds = \frac{3}{2}$$

e

$$\int_{x_{-1}}^{x_2} f(x)dx = h\sum_{k=0}^{1} \Delta^k f_0 c_{1k} + h^2 \int_{-1}^{2} \binom{s}{1} f'(\xi) ds$$
$$= h(c_{10} f_0 + c_{11}(f_1 - f_0)) + h^3 \int_{-1}^{2} \binom{s}{1} f''(\xi) ds$$
$$= h\left(3f_0 + \frac{3}{2}(f_1 - f_0)\right) + \frac{3h^3}{4} f''(\xi)$$
$$= \frac{h}{2}(3f_0 + 3f_1) + \frac{3h^3}{4} f''(\xi)$$

Con $n = 2$, se obtiene la regla

$$\int_{x_{-1}}^{x_3} f(x)dx = \frac{4h}{3}(2f_0 - f_1 + 2f_2) + \frac{14h^5}{45} f^{(4)}(\xi)$$

www.FreeLibros.me

El tipo de reglas obtenidas así, se conoce como las *fórmulas abiertas de Newton-Cotes*. Se tiene, como en el caso de las fórmulas cerradas, el siguiente teorema que nos permite estimar el error.

Teorema 6.2 Para n dado, si n es par y $f \in C^{n+2}[a, b]$, y si n es impar y $f \in C^{n+1}[a, b]$. Definiendo $h = \frac{b-a}{n+2}$ y $x_i = a + ih, i = 0, 1, \ldots, n+1$. Entonces

$$\int_a^b f(x)dx = h\sum_{k=0}^{n} c_{nk}\Delta^k f_0 + \frac{h^{n+2} f^{(n+1)}(\xi)}{(n+1)!}\int_{-1}^{n+1} s(s-1)\cdots(s-n)ds, \ n \text{ impar}$$

$$\int_a^b f(x)dx = h\sum_{k=0}^{n} c_{nk}\Delta^k f_0 + \frac{h^{n+3} f^{(n+2)}(\xi)}{(n+2)!}\int_{-1}^{n+1} s^2(s-1)\cdots(s-n)ds, \ n \text{ par}$$

donde

$$c_{nk} = \int_{-1}^{n+1}\binom{s}{k}ds$$

$$\xi \in (a, b).$$

Observe que cuando n es par, el orden es $O(h^{n+3})$, pero cuando n es impar el orden es simplemente $O(h^{n+2})$, por lo que es aconsejable usar sólo valores pares de n.

6.4.3 Fórmulas compuestas

Las fórmulas de integración anteriores no son adecuadas para intervalos de integración grandes. Para intervalos grandes son necesarias fórmulas de muy alto orden, lo cual las hace inadecuadas. Para obviar este proceso, se propone una alternativa consistente en subdividir el intervalo de integración en subintervalos pequeños y aplicar las fórmulas dadas. Para iniciar esto, se considerarán las fórmulas de los rectángulos por la izquierda y por la derecha, las cuales se obtienen al integrar en el intervalo (x_0, x_1) un polinomio de grado cero basado en los puntos $(x_0, f(x_0))$ y $(x_1, f(x_1))$, respectivamente,

$$\int_{x_0}^{x_1} f(x)dx = hf(x_0) + \frac{h^2}{2}f'(\xi)$$

$$\int_{x_0}^{x_1} f(x)dx = hf(x_1) - \frac{h^2}{2}f'(\xi)$$

Considerando el problema

$$\int_a^b f(x)dx$$

Dado n, se definen $h = \frac{(b-a)}{n}$ y $x_j = a + jh, \ j = 0, 1, \ldots, n$. Se tiene entonces que

$$\int_a^b f(x)dx = \sum_{j=0}^{n-1}\int_{x_j}^{x_{j+1}} f(x)dx$$

Aplicando la regla de los rectángulos por la izquierda en cada subintervalo, se obtiene

$$\int_a^b f(x)dx = \sum_{j=0}^{n-1}\left(hf(x_j) + \frac{h^2}{2}f'(\xi_j)\right) = h\sum_{j=0}^{n-1} f(x_j) + \frac{h^2}{2}\sum_{j=0}^{n-1} f'(\xi_j)$$

Si $f \in C[a, b]$ se tiene por el teorema del valor intermedio que existe ξ en $[a, b]$ tal que

www.FreeLibros.me

$$\sum_{j=0}^{n-1} f'(\xi_j) = nf'(\xi)$$

por lo que

$$\int_a^b f(x)dx = h\sum_{j=0}^{n-1} f(x_j) + \frac{h^2}{2}nf'(\xi)$$

Dado que $nh = b-a$ se obtiene

$$\int_a^b f(x)dx = h\sum_{j=0}^{n-1} f(x_j) + \left(\frac{b-a}{2}\right)f'(\xi)h$$

Esta regla se conoce como la *regla compuesta de los rectángulos por la izquierda*. De manera similar se obtiene la *regla compuesta de los rectángulos por la derecha*

$$\int_a^b f(x)dx = h\sum_{j=1}^{n} f(x_j) - \left(\frac{b-a}{2}\right)f'(\xi)h$$

Cabe hacer notar que estas dos reglas son de orden $O(h)$. Si se considera la regla de los trapecios se tendrá que

$$\begin{aligned}\int_a^b f(x)dx &= \sum_{j=0}^{n-1}\left(\frac{h}{2}(f_j + f_{j+1}) - \frac{h^3}{12}f''(\xi_j)\right)\\ &= \frac{h}{2}\left(f_0 + 2\sum_{j=1}^{n-1} f(x_j) + f_n\right) - \frac{h^3}{12}\sum_{j=0}^{n-1} f''(\xi_j)\\ &= \frac{h}{2}\left(f_0 + 2\sum_{j=1}^{n-1} f(x_j) + f_n\right) - \frac{h^3}{12}nf''(\xi)\\ &= \frac{h}{2}\left(f_0 + 2\sum_{j=1}^{n-1} f(x_j) + f_n\right) - \frac{b-a}{12}f''(\xi)h^2\end{aligned}$$

La *regla de Simpson compuesta* se obtiene al considerar un número par de subintervalos

$$\begin{aligned}\int_a^b f(x)dx &= \sum_{j=1}^{n/2}\int_{x_{2j-2}}^{x_{2j}} f(x)dx\\ &= \sum_{j=1}^{n/2}\left(\frac{h}{3}(f_{2j-2} + 4f_{2j-1} + f_{2j}) - \frac{h^5}{90}f^{(iv)}(\xi_j)\right)\\ &= \frac{h}{3}\sum_{j=1}^{n/2}(f_{2j-2} + 4f_{2j-1} + f_{2j}) - \frac{h^5}{90}\sum_{j=1}^{n/2} f^{(iv)}(\xi_j)\end{aligned}$$

$$\begin{aligned}&= \frac{h}{3}\sum_{j=1}^{n/2}(f_{2j-2} + 4f_{2j-1} + f_{2j}) - \frac{h^5}{90}\frac{n}{2}f^{(iv)}(\xi)\\ &= \frac{h}{3}\sum_{j=1}^{n/2}(f_{2j-2} + 4f_{2j-1} + f_{2j}) - \frac{(b-a)}{180}f^{(iv)}(\xi)h^4\end{aligned}$$

Esto es

$$\int_a^b f(x)dx = \frac{h}{3}(f_0 + 4f_1 + 2f_2 + 4f_3 + \cdots + 4f_{n-3} + 2f_{n-2} + 4f_{n-1} + f_n) - \frac{(b-a)}{180}f^{(iv)}(\xi)h^4$$

www.FreeLibros.me

Para las fórmulas de Newton-Cotes de orden tres, es decir, para la regla de Simpson 3/8 es necesario que el número de subintervalos sea un múltiplo de 3, y se obtiene

$$I_n = \int_{x_0}^{x_n} f(x)dx \approx \frac{3h}{8}(f_0 + 3f_1 + 3f_2 + f_3) + \frac{3h}{8}(f_3 + 3f_4 + 3f_5 + f_6) + \cdots + \frac{3h}{8}(f_{n-3} + 3f_{n-2} + 3f_{n-1} + f_n)$$

$$= \frac{3h}{8}[f_0 + 3f_1 + 3f_2 + 2f_3 + 3f_4 + 3f_5 + 2f_6 + \cdots + f_n]$$

Generalizando la formulación se llega a

$$I_n \approx \frac{3h}{8}\left[f_0 + 3\sum_{\substack{i=1 \\ i\neq 3n}}^{n-1} f_i + 2\sum_{i=3n}^{n-3} f_i + f_n \right]$$

Para el método del punto medio, la regla compuesta se reduce a

$$I_n = \int_{x_0}^{x_n} f(x)dx \approx 2h(f_1) + 2h(f_3) + \cdots + 2h(f_{n-1}), n \text{ par}$$

$$= 2h[f_1 + f_3 + f_5 \cdots + f_{n-1}]$$

La sección 6.8.6 proporciona el código en Matlab para la regla de integración de punto medio.

EJEMPLO 6.4

Considere el problema de evaluar la integral $I = \int_1^4 e^x dx$. Usando la regla de los rectángulos por la derecha compuesta, con $n = 6 \left(h = (4-1)/6 = \frac{1}{2} \right)$, se obtiene

$$I \approx \left(\frac{1}{2}\right)\left(e^1 + e^{\frac{3}{2}} + e^2 + e^{\frac{5}{2}} + e^3 + e^{\frac{7}{2}} \right) = 39.9862549$$

Con el método de los trapecios compuestos y el mismo valor de n

$$I \approx \frac{\frac{1}{2}}{2}\left(e^1 + 2e^{\frac{3}{2}} + 2e^2 + 2e^{\frac{5}{2}} + 2e^3 + 2e^{\frac{7}{2}} + e^4 \right) = 52.956222$$

Usando el método de Simpson se obtiene

$$I \approx \frac{\frac{1}{2}}{3}\left(e^1 + 4e^{\frac{3}{2}} + 4e^2 + 2e^{\frac{5}{2}} + 4e^3 + 4e^{\frac{7}{2}} + e^4 \right) = 51.8973596$$

En la siguiente tabla se muestran los resultados obtenidos con diferentes métodos.

Tabla 6.6 Resultados de la integral, aproximándola con diferentes métodos.

Método	Integral aproximada
Rectángulos por la derecha	39.9862549
Rectángulos por la izquierda	65.926189
Trapecios	52.956222

www.FreeLibros.me

(*Continuación*)

Método	Integral aproximada
Simpson	51.8973596
Simpson 3/8	51.9181166
Punto medio	49.779635

El valor exacto de la integral es $I=\int_1^4 e^x dx = e^4 - e = 51.87986820\ldots$. Se puede observar que aunque se tomó un número pequeño de subintervalos, la regla de Simpson es una buena aproximación al valor de la integral.

6.5 Cuadratura de Gauss

Si se considera la fórmula

$$\int_a^b f(x)dx = \sum_{i=0}^{n} a_i f(x_i) + E$$

hay $2(n+1)$ parámetros que se deben determinar, es decir, a_i y x_i. Debido a que los polinomios de grado $2n+1$ necesitan $2(n+1)$ condiciones para fijar los coeficientes, parece posible que los $2(n+1)$ parámetros se pueden elegir de forma tal que todos los polinomios de menor o igual grado a $2n+1$ se puedan integrar con error cero [Nakamura, 1992], [Maron *et al.*, 1995], [Nieves *et al.*, 2002].

Si se considera el método dado en la sección 6.4.1.1 para encontrar los coeficientes a_i, se puede ver que no es fácil extenderlo para encontrar los nodos x_i. Considerando el caso donde $n=1$, que conduce a las siguientes ecuaciones usando el método de Newton-Cotes

$$\begin{aligned}
f(x)&=1 & 2h&=a_0+a_1\\
f(x)&=x & \frac{4h^2}{2}&=x_0\cdot a_0+x_1\cdot a_1\\
f(x)&=x^2 & \frac{8h^3}{3}&=x_0^2\cdot a_0+x_1^2\cdot a_1\\
f(x)&=x^3 & \frac{16h^3}{4}&=x_0^3\cdot a_0+x_1^3\cdot a_1,
\end{aligned}$$

aunque estas ecuaciones son lineales respecto a los a_i, son no lineales para los x_i y no son fáciles de resolver. La situación se vuelve más complicada para valores grandes de n. Afortunadamente, la teoría matemática asociada a los polinomios ortogonales proporciona un medio de encontrar los nodos y, entonces, los coeficientes a_i se pueden obtener por el método previo.

6.5.1 Polinomios ortogonales

La propiedad fundamental de polinomios ortogonales es, si hay un conjunto de polinomios

$$Q_k(x)(k=0,1,\ldots,n+1)$$

que son ortogonales en el intervalo $[a, b]$, entonces

$$\int_a^b Q_{n+1}(x)S_k(x)dx = 0 \qquad k=0,1,2,\ldots,n$$

www.FreeLibros.me

$S_k(x)$ es cualquier polinomio de grado k. Esta propiedad se cumple para cualquier $S_k(x)$ arbitraria, ya que cualquier polinomio se puede expresar como una combinación lineal de los polinomios ortogonales $Q_k(x)$. Si ahora se toma un polinomio arbitrario $P_{2n+1}(x)$ de grado $2n+1$ y se divide entre $Q_{n+1}(x)$, se obtiene:

$$P_{2n+1}(x) = Q_{n+1}(x) \cdot L_n(x) + R(x)$$

donde $R(x)$ tiene grado máximo n. La integral se puede dividir ahora en dos partes, por ejemplo

$$\int_a^b P_{2n+1}(x)dx = \int_a^b Q_{n+1}(x) \cdot L_n(x)dx + \int_a^b R(x)dx \tag{6.12}$$

y, por la propiedad de ortogonalidad, la primera integral es cero. Así:

$$\int_a^b P_{2n+1}(x)dx = \int_a^b R(x)dx \approx \sum_{i=0}^{n} a_i R(x_i)$$

Si ahora se eligen los puntos $x_i\,(i = 0, 1, ..., n)$ como los ceros de $Q_{n+1}(x)$, entonces se llega a

$$P_{2n+1}(x_i) = R(x_i), \qquad i = 0, 1, 2, \ldots, n$$

Debido a que el primer término de la ecuación (6.12) se vuelve cero, se tiene

$$\int_a^b P_{2n+1}(x)dx = \sum_{i=0}^{n} a_i P_{2n+1}(x_i)$$

donde los nodos x_i ahora se conocen como los ceros de $Q_{n+1}(x)$, y los coeficientes a_i se pueden encontrar por el método de Newton-Cotes.

6.5.2 Pesos en la cuadratura de Gauss

Si se usa la fórmula anterior en el intervalo $[-1, +1]$, entonces los polinomios ortogonales relevantes son los polinomios de Legendre. Pero es posible utilizar otros polinomios ortogonales haciendo una insignificante variación en las condiciones del problema. Primero, hay que notar que una integral con respecto a x entre los límites finitos a y b se pueden transformar como una integral en t con límites $[-1, +1]$ por medio de la transformación

$$t = \frac{2x - (a+b)}{b-a}$$

Debido a esto, los polinomios de Legendre son convenientes para cualquier intervalo finito, con los cambios de variable apropiados. Si se introduce la función de peso $w(x) > 0$ en el integrando, y se usan polinomios que sean ortogonales respecto a esta función de peso, entonces se pueden usar muchos polinomios ortogonales diferentes al escribir la integral en la forma

$$\int_{-1}^{+1} f(x)dx = \int_{-1}^{+1} w(x) \cdot \frac{f(x)}{w(x)}dx$$

Así, si $g(x) = \dfrac{f(x)}{w(x)}$, se tiene

$$\int_{-1}^{+1} w(x) \cdot g(x)dx = \sum_{i=0}^{n} a_i \cdot g(x_i)$$

Como un ejemplo, se consideran los polinomios ortogonales que surgen cuando $w(x) = (1-x^2)^{-\frac{1}{2}}$. Estos polinomios son los polinomios de Tchebyshev, los cuales tienen dos propiedades que aquí resultan de particular interés. Primero, todos los coeficientes a_i tienen el mismo valor $\dfrac{\pi}{(n+1)}$, lo cual reduce lige-

www.FreeLibros.me

ramente las necesidades de cómputo y reduce el error de redondeo. También, los nodos de la fórmula de integración de Tchebyshev están dados por una fórmula sencilla

$$x_j = \cos\left[\frac{(2j+1)\pi}{2(n+1)}\right], \qquad j = 0, 1, 2, \ldots, n$$

Observe que si se calcula el valor de la integral para un valor de n y es necesario incrementar este valor para obtener una mejor precisión, es posible hacer que coincidan algunos valores de nodos en las dos fórmulas y reducir los cálculos extras. Por ejemplo, si se usaran los nodos $\cos\left[\frac{\pi}{6}\right]$, $\cos\left[\frac{3\pi}{6}\right]$ y $\cos\left[\frac{5\pi}{6}\right]$, el grado de la fórmula se puede multiplicar por 3 para dar los puntos $\cos[\theta_i]$ donde los θ_i están dados por $\frac{\pi}{18}, \frac{3\pi}{18}$ y $\frac{5\pi}{18}$. Un tercio de estos valores coinciden con el conjunto anterior y los $f(x_i)$ no necesitan recalcularse. Esta propiedad no la tienen los otros polinomios ortogonales.

6.5.3 Cuadratura de Gauss-Legendre

La fórmula para la aproximación de una integral está definida por

$$\int_{-1}^{+1} f(x)dx = \sum_{i=0}^{n} a_i \cdot f(x_i) + E \tag{6.13}$$

donde los x_i son las raíces del polinomio de Legendre utilizado para aproximar la integral, los cuales tienen la relación de recurrencia

$$nP_n(x) - x(2n-1)P_{n-1}(x) + (n-1)P_{n-2}(x) = 0$$

A continuación se enuncian los primeros cinco polinomios de Legendre

$$P_0(x) = 1$$

$$P_1(x) = x$$

$$P_2(x) = \frac{1}{2}(3x^2 - 1)$$

$$P_3(x) = \frac{1}{2}(5x^3 - 3x)$$

$$P_4(x) = \frac{1}{8}(35x^4 - 30x^2 + 3)$$

Para aproximar la integral por un método de cuadratura de Gauss-Legendre, se utiliza la base ortogonal

$$f(x) = 1, x, x^2, x^3, \ldots, x^n$$

y se utiliza la región de ortogonalidad de los polinomios de Legendre.

6.5.3.1 Cuadratura de Gauss-Legendre de primer orden

Para $n = 1$, se tiene la siguiente aproximación

$$\int_{-1}^{+1} f(x)dx \approx \sum_{i=0}^{1} a_i \cdot f(x_i) = a_0 \cdot f(x_0) + a_1 \cdot f(x_1)$$

www.FreeLibros.me

Debido a que tanto a_i como x_i son incógnitas, se necesitan 4 ecuaciones. Así, el grupo de ecuaciones es

1. Para $f(x)=1$, la función será $f(x_0)=1$ y $f(x_1)=1$. Así se tiene

$$\int_{-1}^{+1} f(x)dx \approx \sum_{i=0}^{1} a_i \cdot f(x_i)$$

$$\int_{-1}^{+1} dx = a_0 \cdot f(x_0)+a_1 \cdot f(x_1)$$

$$x\Big|_{-1}^{+1} = a_0 \cdot (1)+a_1 \cdot (1)$$

$$2 = a_0 + a_1$$

2. Para $f(x)=x$, la función será $f(x_0)=x_0$ y $f(x_1)=x_1$. Así se tiene

$$\int_{-1}^{+1} f(x)dx \approx \sum_{i=0}^{1} a_i \cdot f(x_i)$$

$$\int_{-1}^{+1} xdx = a_0 \cdot f(x_0)+a_1 \cdot f(x_1)$$

$$\left.\frac{x^2}{2}\right|_{-1}^{+1} = a_0 \cdot (x_0)+a_1 \cdot (x_1)$$

$$0 = a_0 \cdot x_0 + a_1 \cdot x_1$$

3. Para $f(x)=x^2$, la función será $f(x_0)=x_0^2$ y $f(x_1)=x_1^2$. Así se tiene

$$\int_{-1}^{+1} f(x)dx \approx \sum_{i=0}^{1} a_i \cdot f(x_i)$$

$$\int_{-1}^{+1} x^2dx = a_0 \cdot f(x_0)+a_1 \cdot f(x_1)$$

$$\left.\frac{x^3}{3}\right|_{-1}^{+1} = a_0 \cdot (x_0^2)+a_1 \cdot (x_1^2)$$

$$\frac{2}{3} = a_0 \cdot x_0^2 + a_1 \cdot x_1^2$$

4. Para $f(x)=x^3$, la función será $f(x_0)=x_0^3$ y $f(x_1)=x_1^3$. Así se tiene

$$\int_{-1}^{+1} f(x)dx = \sum_{i=0}^{1} a_i \cdot f(x_i)$$

$$\int_{-1}^{+1} x^3dx = a_0 \cdot f(x_0)+a_1 \cdot f(x_1)$$

$$\left.\frac{x^4}{4}\right|_{-1}^{+1} = a_0 \cdot (x_0^3)+a_1 \cdot (x_1^3)$$

$$0 = a_0 \cdot x_0^3 + a_1 \cdot x_1^3$$

El grupo de ecuaciones queda de la siguiente forma

www.FreeLibros.me

$$2 = a_0 + a_1$$

$$0 = a_0 \cdot x_0 + a_1 \cdot x_1$$

$$\frac{2}{3} = a_0 \cdot x_0^2 + a_1 \cdot x_1^2$$

$$0 = a_0 \cdot x_0^3 + a_1 \cdot x_1^3$$

Como las incógnitas son x_0 y x_1, se utiliza un polinomio de Legendre de orden 2, $P_2(x) = \frac{1}{2}(3x^2 - 1) = 0$, y se utilizan sus raíces, las cuales son $x_0 = -\frac{1}{\sqrt{3}}$ y $x_1 = +\frac{1}{\sqrt{3}}$. Sustituyendo estos valores en el sistema de ecuaciones, el sistema sólo tiene dos incógnitas, por tanto se puede resolver con dos ecuaciones, las cuales en este caso son

$$2 = a_0 + a_1$$

$$0 = a_0 \cdot \left(-\frac{1}{\sqrt{3}}\right) + a_1 \cdot \left(+\frac{1}{\sqrt{3}}\right)$$

Así se obtienen los valores de $a_0 = 1$ y $a_1 = 1$. Por tanto, la aproximación a la integral queda de la siguiente manera

$$\int_{-1}^{+1} f(x)dx = a_0 \cdot f(x_0) + a_1 \cdot f(x_1) = f(x_0) + f(x_1)$$

Para hacer el cambio a un intervalo general, se tiene que

$$\int_a^b f(z)dz = \int_{-1}^{+1} f(z)\left(\frac{dz}{dx}\right)dx = \frac{b-a}{2}\sum_{k=0}^{N} w_k \cdot f(z_k)$$

donde

$$\frac{dz}{dx} = \frac{b-a}{2}$$

$$z_k = \frac{(b-a)x_k + b + a}{2}$$

$$w_k = a_k$$

La sección 6.8.7 proporciona el código en Matlab para realizar la integración numérica de una función utilizando la técnica de cuadratura de Gauss-Legendre de orden uno.

6.5.3.2 Cuadratura de Gauss-Legendre de segundo orden

Para $n = 2$ se tiene la aproximación

$$\int_{-1}^{+1} f(x)dx \approx \sum_{i=0}^{2} a_i \cdot f(x_i) = a_0 \cdot f(x_0) + a_1 \cdot f(x_1) + a_2 \cdot f(x_2)$$

Como las incógnitas son x_0, x_1 y x_2 se utiliza un polinomio de Legendre de orden 3, $P_3(x) = \frac{1}{2}(3x^3 - 3x) = 0$, y se utilizan sus raíces, las cuales son $x_0 = -\sqrt{\frac{3}{5}}$, $x_1 = 0$ y $x_2 = +\sqrt{\frac{3}{5}}$. Así, el grupo de seis ecuaciones utilizando estas raíces es

$$\text{Para } f(x) = 1,\ 2 = a_0 + a_1 + a_2$$

www.FreeLibros.me

$$\text{Para } f(x)=x, \quad 0=a_0 \cdot x_0+0+a_2 \cdot x_2$$

$$\text{Para } f(x)=x^2, \quad \frac{2}{3}=a_0 \cdot x_0^2+0+a_2 \cdot x_2^2$$

Sustituyendo los valores de las raíces se obtienen los valores de $a_0=\frac{5}{9}$, $a_1=\frac{8}{9}$ y $a_2=\frac{5}{9}$. Así, la aproximación a la integral queda de la siguiente manera

$$\int_{-1}^{+1} f(x)dx=a_0 \cdot f(x_0)+a_1 \cdot f(x_1)+a_2 \cdot f(x_2)=\frac{5}{9} \cdot f(x_0)+\frac{8}{9} \cdot f(x_1)+\frac{5}{9} \cdot f(x_2)$$

La sección 6.8.8 proporciona el código en Matlab para realizar la integración numérica de una función utilizando la técnica de cuadratura de Gauss-Legendre de orden 2.

6.5.3.3 Generalización de las cuadratura de Gauss-Legendre

Analizando el proceso para obtener las formulaciones de cuadratura de Gauss-Legendre, se puede obtener un procedimiento general para cualquier orden. Para una aproximación de α términos, se parte de las raíces del polinomio de Legendre $P^{\alpha}(x)$ y de un sistema de ecuaciones de $\alpha \times \alpha$. Por ejemplo, si se tiene que $\alpha=2$, se llega al sistema de ecuaciones siguiente

$$2=a_0 \cdot x_0^0+a_1 \cdot x_1^0+a_2 \cdot x_2^0$$

$$0=a_0 \cdot x_0^1+a_1 \cdot x_1^1+a_2 \cdot x_2^1$$

$$\frac{2}{3}=a_0 \cdot x_0^2+a_1 \cdot x_1^2+a_2 \cdot x_2^2$$

En forma matricial, se tiene que

$$\begin{bmatrix} x_0^0 & x_1^0 & x_2^0 \\ x_0^1 & x_1^1 & x_2^1 \\ x_0^2 & x_1^2 & x_2^2 \end{bmatrix}\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}=\begin{bmatrix} 2 \\ 0 \\ \frac{2}{3} \end{bmatrix}$$

La forma general de este sistema de ecuaciones es con $\alpha=n$, de donde se obtiene

$$\begin{bmatrix} x_0^0 & x_1^0 & x_2^0 & \cdots & x_n^0 \\ x_0^1 & x_1^1 & x_2^1 & \cdots & x_n^1 \\ x_0^2 & x_1^2 & x_2^2 & \cdots & x_n^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_0^n & x_1^n & x_2^n & \cdots & x_n^n \end{bmatrix}\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}=\begin{bmatrix} 2 \\ 0 \\ \frac{2}{3} \\ \vdots \\ 0 \text{ o } \frac{2}{n+1} \end{bmatrix}$$

De esta manera es fácil notar que la única restricción en el orden de la aproximación es el cálculo de la inversa de la matriz.

EJEMPLO 6.4

Resolver por el método de Gauss-Legendre de orden 1, la siguiente integral

$$\int_0^2 \left(2x^2e^{-3x}\right)dx=\frac{b-a}{2}\sum_{k=0}^{N} w_k \cdot f(z_k)$$

Para una aproximación de orden $n=1$, se tiene que

$$w_0=1 \qquad x_0=-\frac{1}{\sqrt{3}}$$

www.FreeLibros.me

$$w_1 = 1 \qquad x_1 = \frac{1}{\sqrt{3}}$$

Por tanto, los puntos trasladados serán

$$z_0 = \frac{(b-a)x_k + b + a}{2} = \frac{(2-0)\left(-\frac{1}{\sqrt{3}}\right) + 2 + 0}{2} = \frac{\sqrt{3}-1}{\sqrt{3}} = 0.422649730810374$$

$$z_1 = \frac{(b-a)x_k + b + a}{2} = \frac{(2-0)\left(+\frac{1}{\sqrt{3}}\right) + 2 + 0}{2} = \frac{\sqrt{3}+1}{\sqrt{3}} = 1.57735026918963$$

Así, la aproximación a la integral será

$$\int_0^2 \left(2x^2e^{-3x}\right)dx = \frac{b-a}{2}\left[w_0 \cdot f(z_0) + w_1 \cdot f(z_1)\right]$$

Sustituyendo los valores de las variables, se tiene

$$\int_0^2 \left(2x^2e^{-3x}\right)dx = \frac{2-0}{2}\left[f(0.422) + f(1.577)\right] = 0.100537446699836 + 0.043831136367052$$

Por tanto el resultado es

$$\int_0^2 \left(2x^2e^{-3x}\right)dx = 0.144368583066888$$

Para una aproximación de orden $n = 2$, se tiene que

w	x	z	$f(z)$
$w_0 = \frac{5}{9}$	$x_0 = -\sqrt{\frac{3}{5}}$	$z_0 = 0.225403330758517$	$f(z_0) = 0.0516745122383073$
$w_1 = \frac{8}{9}$	$x_1 = 0$	$z_1 = 1$	$f(z_1) = 0.0995741367357279$
$w_2 = \frac{5}{9}$	$x_2 = +\sqrt{\frac{3}{5}}$	$z_2 = 1.77459666924148$	$f(z_2) = 0.0306998813215781$

Por tanto, la aproximación a la integral será

$$\int_0^2 (2x^2e^{-3x})dx = \frac{b-a}{2}[w_0 \cdot f(z_0) + w_1 \cdot f(z_1) + w_2 \cdot f(z_2)]$$

Sustituyendo valores, se tiene

$$\int_0^2 (2x^2e^{-3x})dx = \frac{2-0}{2}\left[\frac{5}{9} \cdot (0.0516745122) + \frac{8}{9} \cdot (0.0995741367) + \frac{5}{9} \cdot (0.0306998813)\right]$$

Así, el valor de la integral es

$$\int_0^2 (2x^2e^{-3x})dx = 0.134273895742806$$

www.FreeLibros.me

EJEMPLO 6.5

Deducir la formulación de cuadratura de Gauss-Legendre de orden 4, para la integral en el intervalo $[a, b]$.

La forma general para obtener los coeficientes para $\alpha = 4$ es

$$\begin{bmatrix} \left(-\frac{1613}{1780}\right)^0 & \left(-\frac{5333}{9904}\right)^0 & (0)^0 & \left(\frac{5333}{9904}\right)^0 & \left(\frac{1613}{1780}\right)^0 \\ \left(-\frac{1613}{1780}\right)^1 & \left(-\frac{5333}{9904}\right)^1 & (0)^1 & \left(\frac{5333}{9904}\right)^1 & \left(\frac{1613}{1780}\right)^1 \\ \left(-\frac{1613}{1780}\right)^2 & \left(-\frac{5333}{9904}\right)^2 & (0)^2 & \left(\frac{5333}{9904}\right)^2 & \left(\frac{1613}{1780}\right)^2 \\ \left(-\frac{1613}{1780}\right)^3 & \left(-\frac{5333}{9904}\right)^3 & (0)^3 & \left(\frac{5333}{9904}\right)^3 & \left(\frac{1613}{1780}\right)^3 \\ \left(-\frac{1613}{1780}\right)^4 & \left(-\frac{5333}{9904}\right)^4 & (0)^4 & \left(\frac{5333}{9904}\right)^4 & \left(\frac{1613}{1780}\right)^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ \frac{2}{3} \\ 0 \\ \frac{2}{5} \end{bmatrix}$$

Por tanto, se tiene que:

$$a_0 = \frac{589}{2\,486}$$

$$a_1 = \frac{1\,075}{2\,246}$$

$$a_2 = \frac{128}{225}$$

$$a_3 = \frac{1\,075}{2\,246}$$

$$a_4 = \frac{589}{2\,486}$$

De esta forma, la integral queda como sigue

$$\int_a^b f(x)dx = \frac{b-a}{2}\left[\frac{589}{2\,486} \cdot f(z_0) + \frac{1\,075}{2\,246} \cdot f(z_1) + \frac{128}{225} \cdot f(z_2) + \frac{1\,075}{2\,246} \cdot f(z_3) + \frac{589}{2\,486} \cdot f(z_4)\right]$$

6.6 Integración de Romberg

Se describe una modificación a la regla trapezoidal compuesta que conduce a una precisión alta, lo cual es muy conveniente para uso computacional [Maron *et al.*, 1995], [Burden *et al.*, 2002], [Rodríguez, 2003]. Para una gran cantidad de funciones, la regla trapezoidal para integrar y su término de error se pueden escribir como sigue

$$\frac{h}{2}[f_0 + 2f_1 + 2f_2 + \cdots + f_m] = I + \sum_{j=1}^{\infty} a_j \cdot h^{2j}$$

debido a esto, existe la posibilidad de utilizar la *extrapolación de Richardson* para mejorar la precisión de los resultados. Si se supone por la simplicidad de notación que el valor inicial de m sea potencia de 2, es decir $m = 2^k$ y, se deja que la aproximación dada por la ecuación anterior esté designada por $T_{0,k}$, por ejemplo:

$$T_{0,k} = I + \sum_{j=1}^{\infty} a_j \cdot h^{2j},$$

la aproximación de la integral se calcula ahora con el intervalo dividido en dos, dando:

www.FreeLibros.me

$$T_{0,\,k+1} = I + \sum_{j=1}^{\infty} a_j \cdot \left(\frac{h}{2}\right)^{2j}$$

Ahora se puede eliminar el primer término de la serie de error tomando una combinación adecuada de estas dos ecuaciones. Así se obtiene

$$4I - I = 4T_{0,\,k+1} - T_{0,\,k} - \sum_{j=2}^{\infty} a_j \cdot \left(\frac{4 \cdot h^{2j}}{2^{2j}} - h^{2j}\right)$$

o bien

$$I = \frac{4T_{0,\,k+1} - T_{0,\,k}}{3} - \sum_{j=2}^{\infty} \frac{a_j \cdot h^{2j}}{3} \cdot \left(\frac{4}{2^{2j}} - 1\right)$$

El primer término del lado derecho de esta ecuación se va a designar como $T_{1,\,k}$ y se verá que el término de error principal ahora es h^4. Sucesivas divisiones entre 2 del intervalo darán una secuencia de valores $T_{0,\,k}$ y cada par sucesivo se puede combinar para dar los valores $T_{1,\,k}$. La secuencia de valores $T_{1,\,k}$ se puede combinar en forma similar para eliminar el término de error h^4 usando la extrapolación de Richardson. Con la fórmula

$$T_{p,\,k} = \frac{1}{4^p - 1}\left(4^p T_{p-1,\,k+1} - T_{p-1,\,k}\right), \qquad k = 0, 1, 2, \ldots, \; p = 1, 2, 3, \ldots,$$

se puede continuar este proceso para formar una sucesión de columnas con términos de error de orden creciente, como se muestra en la tabla 6.7.

Tabla 6.7 Términos de error de orden creciente de la extrapolación de Richarson.

h^2	h^4	h^6	h^8
$T_{0,0}$			
$T_{0,1}$	$T_{1,0}$		
$T_{0,2}$	$T_{1,1}$	$T_{2,0}$	
$T_{0,3}$	$T_{1,2}$	$T_{2,1}$	$T_{3,0}$

Debido a que $h = \frac{(b-a)}{2^k}$, se observa que el término de error para la aproximación $T_{p,\,k}$ es del orden $\left[\left(\frac{b-a}{2^k}\right)\right]^{2p+2}$, con cada columna de valores convergiendo más rápido al valor verdadero de la integral. Este método es muy apropiado para uso computacional, debido a que es posible comparar los valores sucesivos para verificar cuándo converge el proceso. La sección 6.8.9 proporciona el código desarrollado en Matlab para integrar numéricamente una función utilizando la técnica de integración de Romberg.

EJEMPLO 6.6

Usando el método de integración de Romberg crear la tabla se sucesiones de la sec x en el intervalo $\left[0, \frac{\pi}{4}\right]$,

Tabla 6.8 Tabla de integración de Romberg para la sec x.

Número de intervalos	$T_{0,\,k}$	$T_{1,\,k}$	$T_{2,\,k}$	$T_{3,\,k}$
1	0.948059			
2	0.899084	0.882759		

www.FreeLibros.me

(*Continuación*)

Número de intervalos	$T_{0,k}$	$T_{1,k}$	$T_{2,k}$	$T_{3,k}$
4	0.885886	0.881487	0.881402	
8	0.882507	0.881381	0.881374	0.881372

NOTA: Con seis cifras significativas y ocho intervalos, el resultado es 0.881372, lo cual concuerda con el resultado correcto que es 0.881374.

6.7 Comparación de métodos

En el caso de funciones disponibles en forma tabulada, se debe escoger una fórmula que esté basada en nodos equidistantes, debido a que la cuadratura gaussiana tiene interpolaciones para encontrar los valores de la función. También resulta conveniente usar intervalos iguales donde los datos provienen de observaciones experimentales. La elección entre fórmulas de Newton-Cotes de orden alto y fórmulas compuestas de orden bajo va a depender de qué tanto se conocen las derivadas de la función. En el caso donde las derivadas de orden alto disminuyen rápidamente, una fórmula de orden alto será más eficiente, suponiendo que la precisión de la computadora es tal, que los errores de redondeo no predominen. Para problemas donde las derivadas de orden alto pueden ser realmente grandes, serán preferibles reglas compuestas de orden bajo; la regla de Simpson será más precisa que la regla trapezoidal, a menos que la cuarta derivada sea considerablemente mayor que la segunda derivada. El problema más difícil es elegir el paso de integración para asegurar suficiente precisión. En las situaciones donde se llevan a cabo muchas integraciones similares, vale la pena calcular muchos resultados con diferentes pasos de integración, para encontrar la longitud del paso de integración en el que se obtiene suficiente precisión. Si sólo se va a llevar a cabo una integración, entonces serán aceptables necesidades adicionales de computación requeridos, por la Cuadratura de Romberg, debido a que la secuencia de valores indica cuándo se ha obtenido la suficiente precisión, y esto puede mostrar que la integración de Romberg convergirá para cualquier función continua.

Si los valores de la función están disponibles en cualquier valor de nodo, éste es el caso en el que los valores de la función se calculan con computadora, entonces será útil la precisión extra disponible de la cuadratura de Gauss. Estas fórmulas son muy valiosas cuando se necesitan evaluaciones repetitivas de una integral, debido a que la eficiencia computacional es muy importante. En semejante caso, se deben hacer cálculos preliminares para elegir el orden de la fórmula, siendo aceptable el tiempo que se toma para esto. Para un solo cálculo, es muy difícil cerciorarse de la precisión alcanzada, a menos que se obtenga más de un valor de la integral para propósitos de comprobación. Si fuese necesario usar una fórmula de orden alto, todos los nodos serían, en general, nuevos valores, y se necesitarían calcular todos los valores de la función, lo cual no es el caso con fórmulas de igual paso de integración. Así, la fórmula de cuadratura gaussiana no es muy conveniente para evaluar una sola integral.

6.8 Programas desarrollados en Matlab

Esta sección provee los códigos de los programas desarrollados en Matlab para todos los ejercicios propuestos, a continuación se listan todos:

6.8.1. Regla rectangular por la izquierda
6.8.2. Regla rectangular por la derecha
6.8.3. Regla trapezoidal
6.8.4. Integración de Simpson 1/3
6.8.5. Integración de Simpson 3/8
6.8.6. Regla de punto medio
6.8.7. Cuadratura de Gauss-Legendre de dos puntos

www.FreeLibros.me

6.8.8. Cuadratura de Gauss-Legendre de tres puntos
6.8.9. Regla rectangular por la izquierda
6.8.10. Integración de Romberg

6.8.1 Regla rectangular por la izquierda

El método de integración de la regla rectangular por la izquierda, es un método de orden cero que construye un rectángulo partiendo del dato inicial f_0.

Programa principal de la regla rectangular por la izquierda

```
% Programa para integrar una función numéricamente, utiliza la regla rectangular por
% la izquierda. El programa se inicia con un intervalo y va aumentando el número de
% ellos hasta que llega a un resultado en el que dos soluciones consecutivas no sean
% diferentes respecto a una tolerancia especificada.
% La función es x^4 + 2*x + 8 en el intervalo [0,30].
% La solución analítica a esta integral da como resultado 4 861 140
clear all
clc
format long g
% Regla rectangular por izquierda.
a = 0;                  % Límite inferior.
b = 30;                 % Límite superior.
N = 1;                  % Número de intervalos.
h = (b-a)/N;            % Tamaño de cada intervalo.
x = (a:h:b);            % Vector de muestras.
fx = x.^4 + 2.*x + 8;   % Valor de la función en los puntos elegidos.
k = 1;                  % Contador de iteraciones.
Irt(k) = h*(fx(1));     % Primer resultado de la integral con un solo intervalo.
% Reducir el paso de integración hasta la convergencia.
tol = 10;
while tol > 1
    N = 2*N;                         % Duplicar el número de muestras.
    h = (b-a)/N;                     % Determinar el paso de integración.
    x = (a:h:b);                     % Vector de muestras.
    fx = x.^4 + 2.*x + 8;            % Valor de la función en los puntos elegidos.
    Sp = length(fx);                 % Número de muestras.
    k = k+1;                         % Aumenta el contador de iteraciones en 1.
    Irt(k) = h*(sum(fx(1:Sp-1)));    % Integral numérica con N muestras.
    tol = abs(Irt(k)-Irt(k-1));      % Evaluación de la tolerancia.
end
% Muestra en la pantalla todas las aproximaciones.
Irt
```

6.8.2 Regla rectangular por la derecha

El método de integración de la regla rectangular por la derecha, es un método de orden cero que construye un rectángulo partiendo del dato final f_n.

Programa principal de la regla rectangular por la derecha

```
% Programa para integrar una función numéricamente; utiliza la regla rectangular por
% la derecha. El programa se inicia con un intervalo y va aumentando el número de
% ellos hasta que llega a un resultado en el que dos soluciones consecutivas no sean
% diferentes respecto a una tolerancia especificada.
% La función es x^4 + 2*x + 8 en el intervalo [0,30].
% La solución analítica a esta integral da como resultado 4 861 140
clear all
clc
```

www.FreeLibros.me

```
format long g
% Regla rectangular por la derecha.
a = 0;                  % Límite inferior.
b = 30;                 % Límite superior.
N = 1;                  % Número de intervalos.
h = (b-a)/N;            % Tamaño de cada intervalo.
x = (a:h:b);            % Vector de muestras.
fx = x.^4 + 2.*x + 8;   % Valor de la función en los puntos elegidos.
k = 1;                  % Contador de iteraciones.
Irt(k) = h*(fx(2));     % Primer resultado de la integral con un solo intervalo.
% Reducir el paso de integración hasta la convergencia.
tol = 10;
while tol > 1
    N = 2*N;                       % Duplicar el número de muestras.
    h = (b-a)/N;                   % Determinar el paso de integración.
    x = (a:h:b);                   % Vector de las muestras.
    fx = x.^4 + 2.*x + 8;          % Valor de la función en los puntos elegidos.
    Sp = length(fx);               % Número de muestras.
    k = k+1;                       % Aumenta el contador de iteraciones en 1.
    Irt(k) = h*(sum(fx(2:Sp)));    % Integral numérica con N muestras.
    tol = abs(Irt(k)-Irt(k-1));    % Evaluación de la tolerancia.
end
% Muestra en la pantalla todas las aproximaciones.
Irt
```

6.8.3 Regla trapezoidal

El método de integración de la regla trapezoidal traza líneas rectas de punto a punto, con lo cual forma trapecios, y los integra para determinar el área.

Programa principal de la regla trapezoidal

```
% Programa para integrar una función numéricamente, utiliza la regla trapezoidal. El
% programa se inicia con un intervalo y va aumentando el número de ellos hasta que
% llega a un resultado en el que dos soluciones consecutivas no sean diferentes
% respecto a una tolerancia especificada.
% La función es x^4 + 2*x + 8 en el intervalo [0,30].
% La solución analítica a esta integral da como resultado 4 861 140
clear all
clc
format long g
% Regla trapezoidal.
a = 0;                         % Límite inferior.
b = 30;                        % Límite superior.
N = 1;                         % Número de intervalos.
h = (b-a)/N;                   % Tamaño de cada intervalo.
x = (a:h:b);                   % Vector de muestras.
fx = x.^4 + 2.*x + 8;          % Valor de la función en los puntos elegidos.
k = 1;                         % Contador de iteraciones.
Irt(k) = (h/2)*(sum(fx));      % Primer resultado de la integral con un solo intervalo.
% Reducir el paso de integración hasta la convergencia.
tol = 1;
while tol > 1e-3
    N = 2*N;                   % Duplicar el número de muestras.
    h = (b-a)/N;               % Determinar el paso de integración.
    x = (a:h:b);               % Vector de muestras.
    fx = x.^4 + 2.*x + 8;      % Valor de la función en los puntos elegidos.
    Sp = length(fx);           % Número de muestras.
    k = k+1;                   % Aumenta el contador de iteraciones en 1.
    Irt(k) = (h/2)*(fx(1) + 2*sum(fx(2:Sp-1)) + fx(Sp));    % Integral numérica con N
                                                            % muestras.
    tol = abs(Irt(k)-Irt(k-1));                             % Evaluación de la
                                                            % tolerancia.
```

www.FreeLibros.me

```
end
% Muestra en la pantalla todas las aproximaciones.
Irt
```

6.8.4 Integración de Simpson 1/3

El método de integración de Simpson 1/3 se basa en la aproximación de un polinomio de orden 2, es decir, necesita dos intervalos para deducir la formulación. Por tanto el método como tal trabaja con tres puntos.

Programa principal de la integración de Simpson 1/3

```
% Programa para integrar una función numéricamente, utiliza la regla de Simpson un
% tercio. El programa inicia con dos intervalos y va aumentando el número de ellos
% hasta que llega a un resultado en el cual dos soluciones consecutivas no sean
% diferentes respecto a una tolerancia especificada.
% La función es x^4 + 2*x + 8 en el intervalo [0,30].
% La solución analítica a esta integral da como resultado 4 861 140
clear all
clc
format long g
% Regla de Simpson 1/3.
a = 0;                                      % Límite inferior.
b = 30;                                     % Límite superior.
N = 2;                                      % Número de intervalos.
h = (b-a)/N;                                % Tamaño de cada intervalo.
x = (a:h:b);                                % Vector de muestras.
fx = x.^4 + 2.*x + 8;                       % Valor de la función en los puntos elegidos.
k = 1;                                      % Contador de iteraciones.
Is1(k) = (h/3)*(fx(1)+4*fx(2)+fx(3));       % Primer resultado de la integral con un solo
                                            % intervalo.
% Reducir el paso de integración.
tol = 1;
while tol > 1e-1
    N = 2*N;                                % Duplicar el número de muestras.
    h = (b-a)/N;                            % Determinar el paso de integración.
    x = (a:h:b);                            % Vector de muestras.
    fx = x.^4 + 2.*x + 8;                   % Valor de la función en los puntos elegidos.
    Sp = length(fx);                        % Número de muestras.
    k = k+1;                                % Aumenta el contador de iteraciones en 1.
    % Integral numérica con N muestras.
    Is1(k) = (h/3)*(fx(1) + 4*sum(fx(2:2:Sp-1)) + 2*sum(fx(3:2:Sp-2)) + fx(Sp));
    tol = abs(Is1(k)-Is1(k-1));             % Evaluación de la tolerancia.
end
% Muestra en la pantalla todas las aproximaciones.
Is1
```

6.8.5 Integración de Simpson 3/8

El método de integración de Simpson 3/8 se basa en la aproximación de un polinomio de orden 3, es decir, necesita tres intervalos para deducir la formulación. Por tanto, el método como tal funciona con cuatro puntos. En Matlab, cuando se usa el signo % significa que se está haciendo un comentario, adicionalmente las funciones propias de Matlab aparecen sombreadas.

Programa principal de la integración de Simpson 3/8

```
% Programa para integrar una función numéricamente, utiliza la regla de Simpson tres
% octavos. El programa inicia con dos intervalos y va aumentando el número de ellos
% hasta que llega a un resultado en el cual dos soluciones consecutivas no sean
% diferentes respecto a una tolerancia especificada.
```

www.FreeLibros.me

```
% La función es x^4 + 2*x + 8 en el intervalo [0,30].
% La solución analítica a esta integral da como resultado 4 861 140
clear all
clc
format long g
% Regla de Simpson 3/8.
a = 0;                                              % Límite inferior.
b = 30;                                             % Límite superior.
N = 3;                                              % Número de intervalos.
h = (b-a)/N;                                        % Tamaño de cada intervalo.
x = (a:h:b);                                        % Vector de muestras.
fx = x.^4 + 2.*x + 8;                               % Valor de la función en los puntos
                                                    % elegidos.
k = 1;                                              % Contador de iteraciones.
Is3(k) = (3*h/8)*(fx(1)+3*fx(2)+3*fx(3)+fx(4));     % Primer resultado de la integral
                                                    % con un solo intervalo.
% Reducir el paso de integración.
tol = 1;
while tol > 1e-1
    N = 2*N;                        % Duplicar el número de muestras.
    h = (b-a)/N;                    % Determinar el paso de integración.
    x = (a:h:b);                    % Vector de muestras.
    fx = x.^4 + 2.*x + 8;           % Valor de la función en los puntos elegidos.
    Sp = length(fx);                % Número de muestras.
    k = k+1;                        % Aumenta el contador de iteraciones en 1.
    % Integral numérica con N muestras.
    Is3(k) = (3*h/8)*(fx(1) + 3*sum(fx(2:3:Sp-2)) + 3*sum(fx(3:3:Sp-1)) + ...
             2*sum(fx(4:3:Sp-1)) + fx(Sp));
    tol = abs(Is3(k)-Is3(k-1));     % Evaluación de la tolerancia.
end
% Muestra en la pantalla todas las aproximaciones.
Is3
```

6.8.6 Regla de integración de punto medio

El método de integración de punto medio es prácticamente un promedio de la regla de integración por la izquierda y por la derecha.

Programa principal de la regla de integración de punto medio

```
% Programa para integrar una función numéricamente, utiliza la regla de Punto Medio.
% El programa inicia con dos intervalos y va aumentando el número de intervalos hasta
% que llega a un resultado en el cual dos soluciones consecutivas no sean diferentes
% respecto a una tolerancia especificada.
% La función es x^4 + 2*x + 8 en el intervalo [0,30].
% La solución analítica a esta integral da como resultado 4 861 140
clear all
clc
format long g
% Regla de Punto Medio.
a = 0;                              % Límite inferior.
b = 30;                             % Límite superior.
N = 2;                              % Número de intervalos.
h = (b-a)/N;                        % Tamaño de cada intervalo.
x = (a:h:b);                        % Vector de muestras.
P = [1 0 0 2 8];                    % Coeficientes del polinomio a evaluar.
fx = polyval(P,x);                  % Valor de la función en los puntos elegidos.
k = 1;                              % Contador de iteraciones.
Ism(k) = (2*h)*(fx(1)+fx(3));       % Primer resultado de la integral con un solo
                                    % intervalo.
% Reducir el paso de integración.
tol = 10;
while tol > 2
```

www.FreeLibros.me

```
    N = 2*N;                            % Duplicar el número de muestras.
    h = (b-a)/N;                        % Determinar el paso de integración.
    x = (a:h:b);                        % Vector de muestras.
    fx = polyval(P,x);                  % Valor de la función en los puntos elegidos.
    Sp = length(fx);                    % Número de muestras.
    k = k+1;                            % Aumenta el contador de iteraciones en 1.
    Ism(k) = (2*h)*(sum(fx(1:2:Sp)));   % Integral numérica con N muestras.
    tol = abs(Ism(k)-Ism(k-1));         % Evaluación de la tolerancia.
end
% Muestra en la pantalla todas las aproximaciones.
Ism
```

6.8.7 Cuadratura de Gauss-Legendre de dos puntos

El método de integración de la cuadratura de Gauss-Legendre de dos puntos se basa en las raíces del polinomio de Legendre de orden 2, las cuales cumplen con el criterio de ortogonalidad discreta en el intervalo que va desde menos uno hasta uno. El método como tal determina los coeficientes de la formulación final para realizar la integración numérica de una función en un intervalo general.

Programa principal de la cuadratura de Gauss-Legendre de dos puntos

```
% Programa para integrar una función numéricamente, utiliza la fórmula de cuadratura
% de Gauss de 2 puntos. El programa inicia con un intervalo y va aumentando el número
% de ellos hasta que llega a un resultado en el cual dos soluciones consecutivas no
% sean diferentes respecto a una tolerancia especificada.
% La función es x^4 + 2*x + 8 en el intervalo [0,30].
% La solución analítica a esta integral da como resultado 4 861 140
clear all
clc
format long g
% Cuadratura de Gauss de dos puntos.
a = 0;                          % Límite inferior.
b = 30;                         % Límite superior.
N = 2;                          % Número de puntos.
h = (b-a)/(N/2);                % Tamaño del intervalo.
r = [-1/sqrt(3) 1/sqrt(3)];     % Raíces del polinomio de Legendre de orden 2.
x = ((b-a).*r+b+a)/2;           % Nodos.
fx = x.^4 + 2.*x + 8;           % Evaluación de la función en los nodos.
k = 1;                          % Contador de iteraciones.
IGauss2(k) = (h/2)*(sum(fx));   % Primer resultado de la integral con un solo
                                % intervalo.
% Reducir el paso de integración.
tol = 1;
c1=a;
c2=b;
while tol > 1e-2
    N = 2*N;                            % Duplicar el número de muestras.
    h = (c2-c1)/(N/2);                  % Nuevo tamaño del intervalo.
    n = (a:h:b);                        % Determina la posición de los intervalos.
    x = [];                             % Inicia los nodos.
    % Ciclo para determinar los nodos en cada uno de los nuevos intervalos.
    for l = 1:length(n)-1
        c1 = n(l);                      % Límite inferior del intervalo l.
        c2 = n(l+1);                    % Límite superior del intervalo l.
        x = [x ((c2-c1).*r+c2+c1)/2];   % Nodos dentro del intervalo l.
    end
    fx = x.^4 + 2.*x + 8;               % Evalúa la función en todos los nodos.
    Sp = length(fx);                    % Número de muestras.
    k = k+1;                            % Incrementa el contador de iteraciones en 1.
    IGauss2(k) = (h/2)*(sum(fx));       % Integral numérica con Sp nodos.
    tol = abs(IGauss2(k)-IGauss2(k-1)); % Evaluación de la tolerancia.
end
```

www.FreeLibros.me

```
% Muestra en la pantalla todas las aproximaciones.
IGauss2
```

6.8.8 Cuadratura de Gauss-Legendre de tres puntos

El método de integración de la cuadratura de Gauss-Legendre de tres puntos se basa en las raíces del polinomio de Legendre de orden 3, la cuales cumplen con el criterio de ortogonalidad discreta en el intervalo que va desde menos uno hasta uno. El método como tal determina los coeficientes de la formulación final para realizar la integración numérica de una función en un intervalo general.

Programa principal de la cuadratura de Gauss-Legendre de tres puntos

```
% Programa para integrar una función numéricamente, utiliza la fórmula de cuadratura
% de Gauss de 3 puntos. El programa inicia con un intervalo y va aumentando el número
% de ellos hasta que llega a un resultado en el cual dos soluciones consecutivas no
% sean diferentes respecto a una tolerancia especificada.
% La función es x^4 + 2*x + 8 en el intervalo [0,30].
% La solución analítica a esta integral da como resultado 4 861 140
clear all
clc
format long g
% Cuadratura de Gauss de tres puntos.
a = 0;                              % Límite inferior.
b = 30;                             % Límite superior.
N = 3;                              % Número de puntos.
h = (b-a)/(N/3);                    % Tamaño del intervalo.
r = [-sqrt(3/5) 0  sqrt(3/5)];     % Raíces del polinomio de Legendre de orden 3.
x = ((b-a).*r+b+a)/2;               % Nodos.
fx = x.^4 + 2.*x + 8;               % Evaluación de la función en los nodos.
k = 1;                              % Contador de iteraciones.
% Primer resultado de la integral con un solo intervalo.
IGauss3(k) = (h/2)*((5/9)*fx(1)+(8/9)*fx(2)+(5/9)*fx(3));
% Reducir el paso de integración.
tol = 1;
c1=a;
c2=b;
while tol > 1e-2
    N = 2*N;                                % Duplicar el número de muestras.
    h = (c2-c1)/(N/3);                      % Nuevo tamaño del intervalo.
    n = (a:h:b);                            % Determina la posición de los intervalos.
    x = [];                                 % Inicia los nodos.
    % Ciclo para determinar los nodos en cada uno de los nuevos intervalos.
    for l = 1:length(n)-1
        c1 = n(l);                          % Límite inferior del intervalo l.
        c2 = n(l+1);                        % Límite superior del intervalo l.
        x = [x ((c2-c1).*r+c2+c1)/2];      % Nodos dentro del intervalo l.
    end
    fx = x.^4 + 2.*x + 8;                   % Evalúa la función en todos los nodos.
    Sp = length(fx);                        % Número de muestras.
    k = k+1;                                % Incrementa el contador de iteraciones en 1.
    % Integral numérica con Sp nodos
    IGauss3(k) = (h/2)*(((5/9)*sum(fx(1:3:Sp-2))+ ...
                 (8/9)*sum(fx(2:3:Sp-1))+(5/9)*sum(fx(3:3:Sp))));
    tol = abs(IGauss3(k)-IGauss3(k-1));    % Evaluación de la tolerancia.
end
% Muestra en la pantalla todas las aproximaciones.
IGauss3
```

6.8.9 Integración de Romberg

El método de integración de Romberg toma la integración trapezoidal y le da un acomodo diferente, con este acomodo se acelera el proceso de convergencia.

www.FreeLibros.me

Programa principal de la integración de Romberg

```
% Programa para integrar una función numéricamente, utiliza la fórmula de integración
% de Romberg. El programa inicia con un intervalo y después con dos intervalos
% aplicando la regla trapezoidal, se implementa la fórmula de integración de Romberg
% para acelerar la convergencia. El programa se detiene cuando dos soluciones
% consecutivas no sean diferentes respecto a una tolerancia especificada.
% La función es x^4 + 2*x + 8 en el intervalo [0,30].
% La solución analítica a esta integral da como resultado 4 861 140
clc
clear all
format long g
% Regla trapezoidal.
a = 0;                        % Límite inferior.
b = 30;                       % Límite superior.
N = 1;                        % Número de intervalos.
h = (b-a)/N;                  % Tamaño de cada intervalo.
x = (a:h:b);                  % Vector de muestras.
P = [1 0 0 2 8];              % Coeficientes del polinomio a evaluar.
fx = polyval(P,x);            % Valor de la función en los puntos elegidos.
k = 1;                        % Contador de iteraciones.
Ir(k,k) = (h/2)*(sum(fx));    % Primer resultado de la integral con un solo intervalo.
% Reducir el paso de integración hasta la convergencia.
tol = 1;
while tol > 1e-3
    N = 2*N;                  % Duplicar el número de muestras.
    h = (b-a)/N;              % Determinar el paso de integración.
    x = (a:h:b);              % Vector de muestras.
    fx = polyval(P,x);        % Valor de la función en los puntos elegidos.
    Sp = length(fx);          % Número de muestras.
    k = k+1;                  % Aumenta el contador de iteraciones en 1.
    Ir(k,1) = (h/2)*(fx(1) + 2*sum(fx(2:Sp-1)) + fx(Sp));   % Integral numérica con N
                                                              % muestras.
    % Ciclo iterativo para calcular el k-ésimo renglón de la tabla de Romberg.
    for m=2:k
        Ir(k,m) = (1/(4^(m-1)-1))*(4^(m-1)*Ir(k,m-1)-Ir(k-1,m-1));   % Fórmula de
                                                                       % Romberg.
    end
    tol = abs(Ir(k,m-1)-Ir(k,m));   % Evaluación de la tolerancia.
end
% Despliega la tabla de valores dada por la fórmula de Romberg.
Ir
```

Problemas propuestos

6.9.1 Calcular la primera y segunda derivadas por la derecha a partir de los datos dados en la siguiente tabla:

Espacio	0	3	6	9	12	15	18
Altura	0	4.5	9	13	18	22.8	27

6.9.2 Usando derivadas centrales de segundo orden, obtener la primera derivada a partir de los datos dados en la siguiente tabla:

Tiempo	0	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6
Espacio	0	10	21	34	47	59	72	87	115

www.FreeLibros.me

6.9.3 Utilizando rectángulos por la derecha, integrar numéricamente la siguiente expresión utilizando 10 secciones,

$$I = \int_0^{\pi} 2t^2 \operatorname{sen}(5t)\,dt$$

6.9.4 Integrar numéricamente con la técnica de rectángulos por la derecha la expresión que se da a continuación, dividiendo el intervalo en 12 secciones.

$$I = \int_{3\pi}^{4\pi} \cos\left(t^2\right)\tan h(3t)\,dt$$

6.9.5 Integrar numéricamente con la técnica de rectángulos por la derecha la expresión que se da a continuación hasta la convergencia,

$$I = \int_1^2 t^{2.3}\log\left(t^{1.02}\right)dt$$

6.9.6 Integrar numéricamente con la técnica de rectángulos por la derecha la expresión que se da a continuación hasta la convergencia,

$$I = \int_0^{\pi/2} t^4 e^{-2.34t}\cos(t)\,dt$$

6.9.7 Integrar numéricamente con la técnica de rectángulos por la derecha la expresión que se da a continuación hasta la convergencia,

$$I = \int_0^3 \left(t^4 - 2t^3 + t^2 e^{-t}\right)dt$$

6.9.8 Integrar utilizando rectángulos por la izquierda y un paso de integración de $\Delta t = 1e^{-3}$, la expresión dada por:

$$I = \int_1^6 \log\left(t^3\right)\cot h(t)\,dt$$

6.9.9 Utilizando el concepto de integración numérica con rectángulos por la izquierda y dividiendo el intervalo de integración en 1 000 secciones, resolver la expresión dada por:

$$I = \int_0^{\pi/2} \left(3 + \cos h(4t)\right)dt$$

6.9.10 Utilizando el concepto de integración numérica con rectángulos por la izquierda hasta la convergencia, resolver la expresión dada por:

$$I = \int_0^5 \left(5\operatorname{sen} h(2t)e^{-3t}\right)dt$$

6.9.11 Utilizando el concepto de integración numérica con rectángulos por la izquierda hasta la convergencia, resolver la expresión dada por:

$$I = \int_0^5 \left(\ln(1+t)e^{-t}\right)dt$$

www.FreeLibros.me

6.9.12 Utilizando el concepto de integración numérica con rectángulos por la izquierda hasta la convergencia, resolver la expresión dada por:

$$I = \int_{0}^{0.1} (0.5 + \cos(377t))\,dt$$

6.9.13 Utilizando la fórmula de integración numérica dada por el método de los trapecios cuando se divide el intervalo de integración en 500 secciones, encontrar la solución a la expresión dada por:

$$I = \int_{0}^{1} \left(0.2 + e^{-10t}\cos(100t)\right)dt$$

6.9.14 Con el método de los trapecios integrar numéricamente la función tabular dada por los siguientes datos:

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13
$f(t)$	1	4	7	1	9	3	7	2	12	34	21	67	8	0

6.9.15 Utilizando la fórmula de integración numérica dada por el método de los trapecios hasta la convergencia, encontrar la solución a la expresión dada por:

$$I = \int_{0}^{0.1} \left(1 + t^2\cos(1000t)\right)dt$$

6.9.16 Utilizando la fórmula de integración numérica dada por el método de los trapecios hasta la convergencia, encontrar la solución a la expresión dada por:

$$I = \int_{0.1}^{2.3} \left(2t^{2.34} + \frac{1}{t}\right)dt$$

6.9.17 Utilizando la fórmula de integración numérica dada por el método de los trapecios hasta la convergencia, encontrar la solución a la expresión dada por:

$$I = \int_{0}^{0.15} (\log(1+t)\sec(10t))\,dt$$

6.9.18 Utilizando la regla de Simpson 1/3 con un $\Delta t = 0.1$, integrar numéricamente la expresión dada por:

$$I = \int_{0}^{2} t^2 e^{-t}\,dt$$

6.9.19 Con la regla de Simpson 1/3, integrar numéricamente la función tabular dada por:

t	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4
$f(t)$	1	3	6	8	9	12	65	76	13	56	75	76	12	67	92

www.FreeLibros.me

6.9.20 Utilizando la regla de Simpson 1/3, integrar numéricamente hasta la convergencia, la expresión dada por:

$$I=\int_0^2 \left(t^2-2t+10\log(1+t)\right)dt$$

6.9.21 Utilizando la regla de Simpson 1/3, integrar numéricamente hasta la convergencia, la expresión dada por:

$$I=\int_0^{12} \left(t^4-90t^2+237\right)dt$$

6.9.22 Utilizando la regla de Simpson 1/3, integrar numéricamente hasta la convergencia, la expresión dada por:

$$I=\int_0^1 \left(10t^3\cos(50t)e^{-2t}+1\right)dt$$

6.9.23 Con la regla de Simpson 1/3, integrar la función dada por la expresión siguiente con $n=42$.

$$I=\int_0^5 \left(t^2-t-6\right)e^{-7t}dt$$

6.9.24 Utilizando la regla de Simpson 3/8, integrar numéricamente la función dada en forma tabular por:

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$f(t)$	6	1	8	3	9	1	2	7	8	4	8	2	5	8	3	4	6	8	1

6.9.25 Con la regla de Simpson 3/8, integrar la función dada por la expresión siguiente hasta la convergencia.

$$I=\int_0^2 \left(2t^5\cos(23t)e^{-1.28t}+3\right)dt$$

6.9.26 Con la regla de Simpson 3/8, integrar la función dada por la expresión siguiente hasta la convergencia.

$$I=\int_0^3 \left(t^2+e^{-2.3t}-t+7\right)dt$$

6.9.27 Con la regla de Simpson 3/8, integrar la función dada por la expresión siguiente hasta la convergencia.

$$I=\int_0^1 \left(\log_2(1+t)-t^2+3\right)dt$$

6.9.28 Utilizando el método de Gauss-Legendre de primer orden, integrar numéricamente, hasta la convergencia, la siguiente función:

www.FreeLibros.me

$$I=\int_{0}^{3}\left(t^{3}+t-10\right)dt$$

6.9.29 Utilizando el método de Gauss-Legendre de primer orden, integrar numéricamente, hasta la convergencia, la siguiente función:

$$I=\int_{0}^{7}\left(\log_{2}(1+t)e^{-t^{1.2}}\cos(5t)\right)dt$$

6.9.30 Utilizando el método de Gauss-Legendre de primer orden, integrar numéricamente, hasta la convergencia, la siguiente función:

$$I=\int_{0}^{4}t^{3.5}e^{-t^{0.2}}dt$$

6.9.31 Con el método de Gauss-Legendre de orden dos, integrar numéricamente, hasta la convergencia, la función:

$$I=\int_{0}^{1}te^{-3t}dt$$

6.9.32 Con el método de Gauss-Legendre de orden dos, integrar numéricamente, hasta la convergencia, la función:

$$I=\int_{0}^{2}\left(\operatorname{sen}(t)e^{-1.57t}+1\right)dt$$

6.9.33 Con el método de Gauss-Legendre de orden dos, integrar numéricamente, hasta la convergencia, la función:

$$I=\int_{0}^{4}\left(\cos(t)e^{-0.57t}+0.5\right)dt$$

6.9.34 Utilizando el método de integración de Romberg, crear la tabla de sucesiones de la integral

$$I=\int_{0}^{\pi/2}t\cos(2t)dt.$$

6.9.35 Con el método de integración de Romberg crear la sucesión de la integral

$$I=\int_{\pi/4}^{1.5}\tan(t)dt.$$

6.9.36 Con el método de integración de Romberg crear la sucesión de la integral

$$I=\int_{0}^{2}t^{2.63}\cos(1.35t)e^{-0.25t}dt.$$

6.9.37 Con el método de integración de Romberg crear la sucesión de la integral

$$I=\int_{0}^{\pi}127e^{-10t}\cos(377t)dt.$$

www.FreeLibros.me

Capítulo 7

Solución de ecuaciones diferenciales ordinarias

7.1 Introducción

La rapidez de cambio de una variable es frecuente en problemas físicos, de manera que las ecuaciones matemáticas asociadas con este problema, son formuladas en términos de las derivadas de esta variable. Como ejemplo están las ecuaciones de movimiento que involucran la distancia x, la velocidad dx/dt y la aceleración d^2x/dt^2. Aunque algunos problemas se pueden resolver mediante una simple regla analítica, muchas ecuaciones diferenciales requieren para su solución un alto grado de preparación y habilidad en matemáticas. En segundo plano, la amplia relación con los problemas físicos demuestra que hay muchas ecuaciones diferenciales para las cuales la solución no se puede representar en una forma matemática sencilla, por lo que los métodos empleados sólo son posibles aproximaciones de la solución.

Sin embargo, hay dificultades asociadas con el uso de los métodos numéricos, ya que una aproximación numérica no es una alternativa rápida si el análisis matemático puede dar una fórmula explícita para la solución. Por otro lado, si el problema no tiene una solución explícita sería útil realizar algún análisis matemático con el objetivo de abordar el problema en la forma más apropiada para el cálculo numérico [Allen *et al.*, 1998], [Chapra *et al.*, 2007], [Nagle *et al.*, 2005], [Zill *et al.*, 2006].

El proceso de integración introduce constantes arbitrarias que son determinadas mediante condiciones iniciales dadas sobre la función o sobre sus derivadas. De acuerdo con la manera en que se especifican las condiciones, surgen dos tipos de problemas: si todas las condiciones necesarias están dadas en un punto único, se tiene un problema de valor inicial; así, el método de solución inicia en el punto conocido y se mueve paso a paso a lo largo del rango de integración. Sin embargo, si las condiciones están dadas en más de un punto, entonces no hay suficiente información para comenzar los cálculos en un único punto, y el método para calcular la solución necesita un conjunto de ecuaciones simultáneas, o el uso de valores estimados en un punto. Este segundo tipo se conoce como un *problema de valor en la frontera*.

Se llama ecuación diferencial a una ecuación que relaciona la variable independiente t, la función incógnita $y = y(t)$ y sus derivadas $y', y'', \ldots, y^{(n)}$; es decir, una ecuación de la forma

$$F\left(t, y, y', y'', \ldots, y^{(n)}\right) = 0 \tag{7.1}$$

www.FreeLibros.me

En otras palabras, se llama *ecuación diferencial* a la que contiene la *derivada* o *diferencial de la función incógnita*. El *orden de una ecuación diferencial* se define como el orden de la derivada más alta de la ecuación. Si la función incógnita $y = y(t)$ depende de una sola variable independiente t, la ecuación se llama *ecuación diferencial ordinaria*. Recibe el nombre de *solución de la ecuación diferencial* ordinaria la función $y = \varphi(t)$ determinada en el intervalo $[a, b]$ junto con sus derivadas sucesivas, hasta el *n-ésimo orden,* inclusive, tal que al hacer la sustitución $y = \varphi(t)$ en la ecuación diferencial ordinaria, ésta se convierte en una identidad con respecto a t en el intervalo $[a, b]$. En general, la solución de una ecuación diferencial ordinaria, si ésta existe, puede no ser única. A fin de tener unicidad, se introducen condiciones extras en las soluciones. Estas condiciones pueden ser de la forma

$$
\begin{aligned}
y(a) &= y_0 \\
y'(a) &= y_0' \\
&\vdots \\
y^{(n-1)}(a) &= y_0^{(n-1)}
\end{aligned}
\tag{7.2}
$$

Estas condiciones se conocen como *condiciones iniciales* para la ecuación diferencial ordinaria (7.1). Las condiciones también se pueden definir en los extremos del intervalo, y entonces se estaría hablando de un problema de valores en la frontera. Las ecuaciones (7.1) y (7.2) se llaman de manera conjunta un *problema de valor inicial.*

EJEMPLO 7.1

La solución general de $y' = ky$, donde k es una constante dada, es

$$y(t) = ce^{kt}$$

donde c es una constante arbitraria. Si, por ejemplo, se da la condición extra de que

$$y(0) = y_0,$$

entonces, tomando $t = 0$, se tiene que $c = y_0$. La única solución que satisface es

$$y(t) = y_0 e^{kt}$$

La descripción de muchos fenómenos físicos se modela matemáticamente con una ecuación diferencial ordinaria. Por ejemplo, el crecimiento no inhibido de bacterias, la descarga de un condensador, el escape de un gas bajo la presión de un contenedor o el decaimiento radioactivo. En todos estos ejemplos, el tiempo se representa por t. La ecuación $y(0) = y_0$ muestra la condición inicial de la magnitud que se está midiendo, por ejemplo, el volumen de un cultivo de bacterias en el tiempo $t = 0$.

En esta sección se analizan los métodos numéricos para resolver (7.1) sujeta a (7.2). Para la solución numérica, el objetivo es buscar una aproximación de los valores numéricos de $y(t)$ para valores particulares de t o una función, por ejemplo un polinomio, que aproxima a y sobre un rango de valores de t. Solamente en raras ocasiones se busca una solución explícita, como en el ejemplo 7.1.

Para iniciar, se considera el problema de valor inicial de primer orden

$$
\begin{aligned}
y' &= f(t, y), t \in [a, b] \\
y(a) &= y_0
\end{aligned}
\tag{7.3}
$$

En la mayoría de los problemas, $f(t, y)$ será complicada, de modo que obtener una solución explícita será una tarea muy laboriosa, o imposible. Esto obliga a usar métodos numéricos. También se abordan *sistemas* de ecuaciones diferenciales ordinarias y ecuaciones diferenciales de *mayor orden.* Los métodos para ambos casos serán extensiones naturales de aquellos usados para resolver una ecuación diferencial de primer orden. La primera interrogante acerca de (7.3) es si existe una solución, o qué condiciones se deben imponer sobre $f(t, y)$ para asegurar su existencia. El siguiente ejemplo ilustra estas ideas.

www.FreeLibros.me

EJEMPLO 7.2

Para $0 \leq t \leq 2$, la solución general de $y' = y^2$, sujeta a la condición inicial $y(0)=1$ es

$$y(t) = \frac{-1}{(t+c)}$$

donde *c* es arbitraria. La condición inicial implica que $c=-1$, y se tiene que

$$y(t) = \frac{-1}{(t-1)}$$

Esta solución no está definida para $t=1$ y, por tanto, el problema de valor inicial *no tiene solución* en [0, 2].

En el ejemplo 7.2, la función $f(t,y)$ es una *función bien comportada*. Obviamente la continuidad de $f(t,y)$ no es suficiente para asegurar la existencia de una solución única, pero la condición de Lipschitz sobre $f(t,y)$ con respecto de *y* sí es suficiente, y el siguiente teorema lo demuestra.

Teorema 7.1 La ecuación diferencial de primer orden $y' = f(t,y)$, sujeta a la condición inicial $y(t_0) = y_0$, donde $t_0 \in [a,b]$, tiene solución única *y* en [*a*, *b*] si $f(t,y)$ es continua para toda *t* en [*a*, *b*] y existe *L*, independiente de *t* tal que

$$|f(t,y) - f(t,z)| \leq L|y-z|$$

para toda $t \in [a, b]$ y todo real *y* y *z*.

La existencia y unicidad de la solución está garantizada por el teorema anterior, pero existe otra cuestión que se debe considerar para determinar el buen comportamiento de la ecuación diferencial en el sentido de que pequeños cambios en la ecuación o en la condición inicial generen pequeños cambios en la solución del problema original. Esto se conoce como un *problema bien planteado*. Se define cuando una ecuación diferencial representa un problema bien planteado como

Definición 7.1 El problema de valor inicial

$$y' = f(t,y), \quad a \leq t \leq b, \quad y(a) = \alpha$$

se dice que es un *problema bien planteado* si

- El problema tiene solución única;
- Para cualquier $\varepsilon > 0$, existe una constante positiva $k(\varepsilon)$, tal que siempre que $|\varepsilon_0| < \varepsilon$ y $\delta(t)$ es continua con $|\delta(t)| < \varepsilon$, y entonces existe una solución única a

  $$z' = f(t,z) + \delta(t), \quad a \leq t \leq b, \quad z(a) = \alpha + \varepsilon_0$$

  y satisface

  $$|z(t) - y(t)| < k(\varepsilon)\varepsilon$$

  para toda $a \leq t \leq b$.

El siguiente teorema establece las condiciones para las cuales un problema con solución única es un problema bien planteado:

www.FreeLibros.me

Teorema 7.2 Sea $D=\{(t,y)\mid a\le t\le b,\ -\infty<y<\infty\}$, si f es continua y existe L, independiente de t tal que

$$|f(t,y)-f(t,z)|\le L|y-z|$$

para todo $(t,y),(t,z)$ en D, entonces el problema de valor inicial

$$y'=f(t,y),\quad a\le t\le b,\quad y(a)=\alpha$$

es un problema bien planteado.

7.2 Métodos de un paso para la solución de ecuaciones diferenciales ordinarias

La solución numérica del problema de valor inicial

$$y'=f(t,y),\ \text{sujeta a}\ y(t_0)=y_0 \tag{7.4}$$

donde $t\in[t_0,b]$ y f satisface las condiciones de los teoremas 7.1 y 7.2 de modo que se garantiza la existencia de una solución única, suponiendo, además, que f es suficientemente diferenciable, esto es, que todas las derivadas usadas en el análisis existen, se busca en una sucesión $y_0,\ y_1,\ y_2,\ldots$, cuyos elementos son una aproximación a $y(t_0),\ y(t_1),\ y(t_2),\ldots$ y $t_0,\ t_1,\ t_2$ en el intervalo $[t_0,b]$.

Se inicia con los métodos conocidos como métodos de un paso para determinar $\{y_k\}$ con el cual se obtiene y_{n+1} solamente a partir de y_n; no se usa $y_{n-1},\ y_{n-2},\ldots$, en forma explícita. El tipo de método que se presenta es sencillo en su derivación y su conceptualización. Este método se conoce como el *método de la serie de Taylor* o, simplemente, *método de Taylor* [Burden *et al.,* 2002], [Cheney *et al.*, 2008], [Maron *et al.*, 1995], [Mathews, 1992].

7.2.1 Serie de Taylor y método de la serie de Taylor

Suponiendo que la función f en (7.4) es *suficientemente* diferenciable, se desea determinar una sucesión $y_0,\ y_1,\ y_2,\ldots$, cuyos elementos son aproximaciones a $y(t_0),\ y(t_1),\ y(t_2),\ldots$, con $t_k=t_0+kh$. Una aproximación es *considerar* la serie de Taylor para y con residuo [Wyley, 1982], [Nakamura, 1992], [Maron *et al.*, 1995], [Mathews *et al.*, 2000], [Burden *et al.*, 2002], [Nieves *et al.*, 2002]. Así se tiene

$$\begin{aligned} y(t_{n+1}) &= y(t_n)+hy'(t_n)+\frac{h^2}{2!}y''(t_n)+\frac{h^3}{3!}y'''(t_n)+\frac{h^4}{4!}y^{(iv)}(t_n)+\cdots \\ &+\frac{h^p}{p!}y^{(p)}(t_n)+h^{p+1}E_{p+1}(t_n) \end{aligned} \tag{7.5}$$

donde

$$E_{p+1}(t_n)=\frac{1}{(p+1)!}y^{(p+1)}(\xi),\qquad t_n<\xi<t_{n+1}$$

Se pueden determinar las derivadas de y a partir de f y de sus derivadas parciales, por ejemplo, para $y'(t)=f(t,y(t))$. Usando la *regla de la cadena* para derivadas parciales, se obtiene

$$\begin{aligned} y''(t) &= \frac{\partial f}{\partial t}+\frac{\partial f}{\partial y}\frac{dy}{dt} \\ &= f_t+f_y y' \\ &= f_t+f_y f \end{aligned}$$

Si se definen las funciones $f^{(r)}(t,\ y)$ recursivamente como

www.FreeLibros.me

$$f^{(r)}(t,y)=f_t^{(r-1)}(t,y)+f_y^{(r-1)}(t,y)f(t,y)$$

para $r\geq 1$, con $f^{(0)}(t,y)=f(t,y)$, entonces

$$y^{(r+1)}(t)=f^{(r)}(t,y) \tag{7.6}$$

para $r\geq 0$.

El resultado es cierto para $r=0$. Diferenciando (7.6) se tiene que

$$\begin{aligned} y^{(r+2)}(t)&=f_t^{(r)}(t,y)+f_y^{(r)}(t,y)y'(t)=f_t^{(r)}(t,y)+f_y^{(r)}(t,y)f(t,y(t))\\ &=f^{(r+1)}(t,y)\end{aligned}$$

lo cual demuestra que se puede extender el resultado a la $(r+2)$-ésima derivada de *y*. Se nota además, que $f^{(r)}(t,y)$ no es la derivada de *f* en el sentido *usual*. Si se sustituye (7.6) en (7.5), se obtiene

$$\begin{aligned} y(t_{n+1})&=y(t_n)+hf(t_n,y(t_n))+\frac{h^2}{2!}f^{(1)}(t_n,y(t_n))+\\ &+\frac{h^3}{3!}f^{(2)}(t_n,y(t_n))+\cdots+\frac{h^p}{p!}f^{(p-1)}(t_n,y(t_n))+h^{p+1}E_{p+1}(t_n)\end{aligned}$$

La aproximación que se obtiene al no considerar el término del residuo es

$$y_{n+1}=y_n+hf(t_n,y_n)+\frac{h^2}{2!}f^{(1)}(t_n,y_n)+\frac{h^3}{3!}f^{(2)}(t_n,y_n)+\cdots+\frac{h^p}{p!}f^{(p-1)}(t_n,y_n) \tag{7.7}$$

y el método de la serie de Taylor consiste en usar esta relación de recurrencia para $n=1,2,\ldots$ junto con $y(t_0)=y_0$. La desventaja de este método es que $f^{(r)}(t,y)$ no es fácil de calcular. Por ejemplo

$$\begin{aligned} f^{(1)}(t,y)&=f_t+f_yf\\ f^{(2)}(t,y)&=f_t^{(1)}+f_y^{(1)}f\\ &=f_{tt}+2f_{ty}f+f_{yy}f^2+(f_t+f_yf)f_y \end{aligned} \tag{7.8}$$

Para $p=1$ se obtiene el famoso *método de Euler*.

$$\begin{aligned} y_{n+1}&=y_n+hf(t_n,y_n),\ n=0,1,2,\ldots\\ y_0&=y(t_0)\end{aligned} \tag{7.9}$$

En la mayoría de los problemas, la precisión del método se incrementa conforme crece *p*, dado que el residuo tiende a cero si la derivada $p+1$ de *y* es continua. Para ilustrar el funcionamiento del método, se considerará un ejemplo concreto.

EJEMPLO 7.3

Considerando el problema de valor inicial

$$y'=-\frac{t}{y},\qquad 0\leq t\leq 5$$

sujeto a la condición $y(0)=5$, las derivadas que aparecen en (7.7) se pueden calcular a partir de la ecuación diferencial incluida en la ecuación (7.8). Así se obtiene lo siguiente:

$$y'=-\frac{t}{y}$$

www.FreeLibros.me

$$y'' = -\frac{y - ty'}{y^2}$$

$$y''' = -\frac{-2yy' + 2t(y')^2 - tyy''}{y^3}$$

$$y^{(iv)} = -\frac{6y(y')^2 - 3y^2y'' - 6t(y')^3 + 6tyy'y'' - ty^2y'''}{y^4}$$

En este punto se trunca y se decide usar sólo los términos hasta incluir h^4 en la fórmula (7.7). El término que no se ha incluido contiene un factor en h^5 y constituye el *error de truncamiento* inherente a este procedimiento. Se dice que el método resultante es de cuarto orden. (*El orden del método de la serie de Taylor es p si se utilizan términos hasta e incluyendo el denotado por* $h^p y^{(p)}(t)/p!$) Se pueden llevar a cabo varias sustituciones con el fin de obtener fórmulas para $y'', y''', \ldots$, que no contengan derivadas de y en el lado derecho de la expresión. Esto no es necesario si las fórmulas se utilizan en el orden presentado. Por su propia naturaleza son recursivas. Pero si se desea, se obtendría

$$y' = -\frac{t}{y}$$

$$y'' = -\frac{y^2 + t^2}{y^3}$$

$$y''' = -\frac{3t(y^2 + t^2)}{y^5}$$

$$y^{(iv)} = -\frac{3(6t^2y^2 + y^4 + 5t^4)}{y^7}$$

Si la función f depende sólo de una variable y no es complicada, el método de la serie de Taylor se puede calcular en forma fácil, como se demuestra en el siguiente ejemplo.

EJEMPLO 7.4

Considerando el problema $y' = y$ con la condición inicial $y(0) = 1$, se tiene que

$$f(t, y) = y$$

$$f^{(1)}(t, y) = y$$

$$f^{(2)}(t, y) = y$$

$$f^{(3)}(t, y) = y$$

y, en general, $f^{(r)}(t, y) = y$.

De (7.7), el método de la serie de Taylor está dado por

$$y_{n+1} = y_n + hy_n + \frac{h^2}{2!}y_n + \cdots + \frac{h^p}{p!}y_n = \left(1 + h + \frac{h^2}{2!} + \cdots + \frac{h^p}{p!}\right)y_n$$

Dado que $y(0) = 1$, se sigue que

www.FreeLibros.me

$$y_n = \left(1 + h + \frac{h^2}{2!} + \cdots + \frac{h^p}{p!}\right)^n$$

El siguiente ejemplo demuestra que no siempre es fácil de calcular el método de la serie de Taylor, aun cuando la función f dependa de una sola variable.

EJEMPLO 7.5

Para la ecuación

$$y' = \frac{1}{1+y^2}, y(0) = 1$$

se tiene que con

$$p = 3$$

y

$$y_0 = 1,$$

con el método de la serie de Taylor,

$$y_{n+1} = y_n + \frac{h}{1+y_n^2} + \frac{h^2}{2}\frac{(-2y_n)}{\left(1+y_n^2\right)^3} + \frac{h^3}{6}\frac{2\left(5y_n^2-1\right)}{\left(1+y_n^2\right)^5}$$

Como se ha observado en los ejemplos anteriores, la principal complicación del método de la serie de Taylor es el cálculo de las derivadas. Por tanto, se propone un método de un paso en el cual no intervengan esos cálculos.

7.2.2 Métodos de Euler

Los métodos de Euler se derivan partiendo del concepto básico de derivada numérica. Según el modo de realizar esta derivada, el método tiene un esquema diferente. A continuación se describen los dos más usados y de deducción más sencilla.

7.2.2.1 Método de Euler-Cauchy

Para ilustra el concepto básico, una aproximación simple por diferencias hacia delante de una derivada se expresa de la siguiente manera [Nakamura, 1992], [Maron *et al.*, 1995], [Mathews *et al.*, 2000], [Burden *et al.*, 2002], [Nieves *et al.*, 2002], [Rodríguez, 2003], [Cordero, 2006]:

$$\frac{u_{j+1}-u_j}{h} = f\left(x_j, u_j\right) = y_n' \qquad j = 0, 1, 2, \ldots, N-1 \tag{7.10}$$

Por simplicidad se escoge una red uniforme

$$x_j = x_0 + jh \qquad j = 0, 1, 2, \ldots, N \text{ y } h = \frac{b-a}{N}$$

La condición inicial es

$$u_0 = y_0 + e_0$$

www.FreeLibros.me

Este método se conoce como de Euler-Cauchy, también llamado *método del polígono* o *regla trapezoidal*. La solución numérica de ecuaciones diferenciales ordinarias por este método equivale a reemplazar las variables que no están bajo el signo de diferenciación por sus valores promedio, y a las diferenciales por diferencias finitas. El programa desarrollado en Matlab se proporciona en la sección 7.9.1.

7.2.2.2 Método de Euler hacia adelante

La existencia de una solución única u_j de la ecuación diferencial lleva a

$$u_{j+1} = u_j + h\, f(x_j, u_j),\ j = 0, 1, 2, \ldots, N-1$$

Por tanto, el método de Euler hacia adelante para la ecuación $y' = f(y, t)$ se obtiene reescribiendo la aproximación por diferencias hacia adelante como

$$\frac{y_{n+1} - y_n}{h} \cong y'_n$$

Despejando y_{n+1}, se llega a

$$y_{n+1} = y_n + h\, y'_n$$

Por tanto, finalmente se obtiene

$$y_{n+1} = y_n + h\, f(y_n, t_n)$$

Para el caso de una ecuación cuyas variables son (y, t) con condiciones iniciales (y_0, t_0) se obtiene el esquema numérico

$$y_1 = y_0 + h\, f(y_0, t_0)$$

$$y_2 = y_1 + h\, f(y_1, t_1)$$

$$\vdots$$

$$y_n = y_{n-1} + h\, f(y_{n-1}, t_{n-1})$$

La sección 7.9.2 proporciona el código computacional desarrollado en Matlab del método de Euler. En este caso específico, el código se implementa para resolver una ecuación en particular; sin embargo, siguiendo el mismo procedimiento, se puede implementar para resolver cualquier ecuación diferencial ordinaria.

EJEMPLO 7.6

Utilizando el método Euler-Cauchy (regla trapezoidal), resolver la ecuación $v' + 4v = e^{-2t}$, en el intervalo $0 \le t \le 1$, con un paso de $h = 0.1$. La condición inicial es $v(0) = 1$.

Discretizando la ecuación como lo indica el método de Euler-Cauchy, se obtiene

$$\frac{v^{n+1} - v^n}{\Delta t} + 4\frac{v^{n+1} + v^n}{2} = \frac{e^{-2t^{n+1}} + e^{-2t^n}}{2}$$

Despejando la variable v^{n+1} se llega a la ecuación

$$v^{n+1} = \frac{e^{-2t^{n+1}} + e^{-2t^n}}{2\left((\Delta t)^{-1} + 2\right)} + \frac{\left((\Delta t)^{-1} - 2\right)}{\left((\Delta t)^{-1} + 2\right)} v^n$$

Si se tiene como condición inicial que

www.FreeLibros.me

$$t_0 = 0 \qquad v_0 = 1$$

entonces la solución para el resto de los puntos es

$$t_1 = 0.1 \qquad v_1 = f\left(v^n, t^n, t^{n+1}\right) = 0.7424$$

$$t_2 = 0.2 \qquad v_2 = f\left(v^n, t^n, t^{n+1}\right) = 0.5570$$

$$t_3 = 0.3 \qquad v_3 = f\left(v^n, t^n, t^{n+1}\right) = 0.4221$$

$$t_4 = 0.4 \qquad v_4 = f\left(v^n, t^n, t^{n+1}\right) = 0.3230$$

$$t_5 = 0.5 \qquad v_5 = f\left(v^n, t^n, t^{n+1}\right) = 0.2494$$

$$t_6 = 0.6 \qquad v_6 = f\left(v^n, t^n, t^{n+1}\right) = 0.1941$$

$$t_7 = 0.7 \qquad v_6 = f\left(v^n, t^n, t^{n+1}\right) = 0.1523$$

$$t_8 = 0.8 \qquad v_6 = f\left(v^n, t^n, t^{n+1}\right) = 0.1202$$

$$t_9 = 0.9 \qquad v_6 = f\left(v^n, t^n, t^{n+1}\right) = 0.0954$$

$$t_{10} = 1.0 \qquad v_6 = f\left(v^n, t^n, t^{n+1}\right) = 0.0761$$

La solución analítica a esta ecuación diferencial es

$$v(t) = \tfrac{1}{2}e^{-2t} + \tfrac{1}{2}e^{-4t}$$

La figura 7.1 muestra la solución numérica comparada con la solución analítica

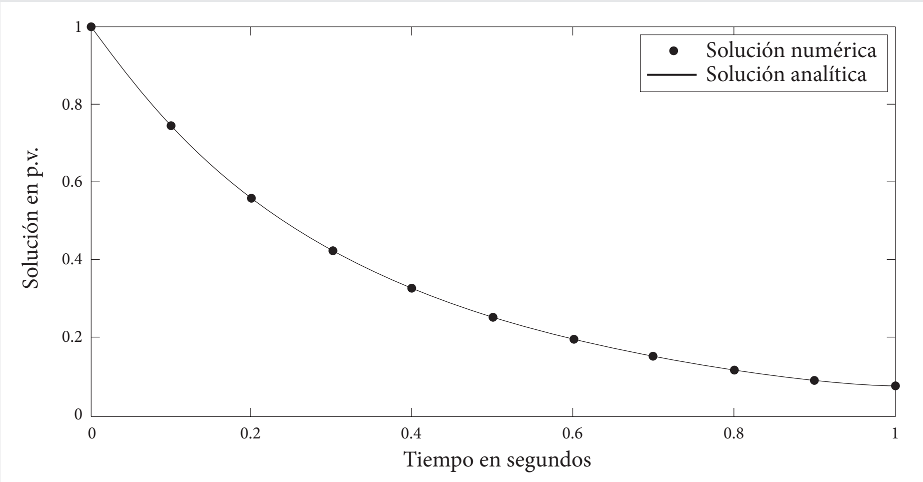


Figura 7.1 Solución numérica *vs*. solución analítica.

EJEMPLO 7.7

Resolver la siguiente ecuación usando el método de Euler en el rango $0 \le x \le 2$ con un paso de $h = 0.5$.

$$\frac{dy}{dx} - y = 0$$

www.FreeLibros.me

La condición inicial es $y(0)=1$. Por tanto, tomando la ecuación y despejando la derivada, se obtiene $y'=y$, es decir, $f(y,x)=y$. El esquema general del método se expresa como

$$y_n = y_{n-1} + h\, f(y_{n-1}, x_{n-1})$$

Por tanto, el primer paso queda de la siguiente forma

$$y_1 = y_0 + h\, f(y_0, x_0)$$

Sustituyendo los valores numéricamente, se tiene que

$$y_1 = y_0 + h\, y_0 = 1 + (0.5)(1) = 1.5$$

Los siguientes pasos hasta completar el intervalo quedan de la siguiente manera

$$y_2 = y_1 + h\, y_1 = 1.5 + (0.5)(1.5) = 2.25$$

$$y_3 = y_2 + h\, y_2 = 2.25 + (0.5)(2.25) = 3.375$$

$$y_4 = y_3 + h\, y_3 = 3.375 + (0.5)(3.375) = 5.062$$

EJEMPLO 7.8

Resolver $y'=-20y+7e^{-0.5t}$ con $y(0)=5$ por medio del método de Euler hacia adelante con $h=0.01$ para $0<t\leq 0.1$.

El esquema numérico para esta ecuación queda de la siguiente manera

$$y_1 = y_0 + h\, y_0' = y_0 + h\left[-20y_0 + 7e^{-0.5t_0}\right]$$

La condición inicial se expresa como

$$t_0 = 0.00 \qquad y_0 = 5$$

De tal forma que los pasos sucesivos hasta completar el intervalo son

$$t_1 = 0.01 \qquad y_1 = y_0 + h\left[-20y_0 + 7e^{-0.5t_0}\right] = 5 + (0.01)\left[-20(5) + 7e^{-0.5(0.00)}\right] = 4.07$$

$$t_2 = 0.02 \qquad y_2 = y_1 + h\left[-20y_1 + 7e^{-0.5t_1}\right] = 4.07 + (0.01)\left[-20(4.07) + 7e^{-0.5(0.01)}\right] = 3.32$$

$$t_3 = 0.03 \qquad y_3 = y_2 + h\left[-20y_2 + 7e^{-0.5t_2}\right] = 2.72982$$

$$t_4 = 0.04 \qquad y_4 = y_3 + h\left[-20y_3 + 7e^{-0.5t_3}\right] = 2.25282$$

$$t_5 = 0.05 \qquad y_5 = y_4 + h\left[-20y_4 + 7e^{-0.5t_4}\right] = 1.87087$$

$$t_6 = 0.06 \qquad y_6 = y_5 + h\left[-20y_5 + 7e^{-0.5t_5}\right] = 1.56497$$

$$t_7 = 0.07 \qquad y_7 = y_6 + h\left[-20y_6 + 7e^{-0.5t_6}\right] = 1.31990$$

$$t_8 = 0.08 \qquad y_8 = y_7 + h\left[-20y_7 + 7e^{-0.5t_7}\right] = 1.12352$$

$$t_9 = 0.09 \qquad y_9 = y_8 + h\left[-20y_8 + 7e^{-0.5t_8}\right] = 0.90607$$

$$t_{10} = 0.1 \qquad y_{10} = y_9 + h\left[-20y_9 + 7e^{-0.5t_9}\right] = 0.83977$$

www.FreeLibros.me

EJEMPLO 7.9

Por medio del método de Euler y con $h = 0.5$ determinar valores de $y(1)$ y $y'(1)$ para $y''(t) - 0.05y'(t) + 0.15y(t) = 0$.

Las condiciones iniciales de este problema son

$$y(0) = 1 \qquad y_0 = 1$$

$$y'(0) = 0 \qquad y'_0 = 0$$

Si se hace la transformación

$$y' = z \qquad y_0 = 1$$

$$z' = 0.05z - 0.15y \qquad z_0 = 0$$

entonces, en $t_1 = 0.5$ se tiene

$$y_1 = y_0 + h y'_0 = 1 + 0.5(0) = 1$$

$$z_1 = z_0 + h z'_0 = 0 + 0.5[0.05(0) - 0.15(1)] = -0.075$$

Para $t_2 = 1$ se obtiene

$$y_2 = y_1 + h y'_1 = 1 + 0.5(-0.075) = 0.96250$$

$$z_2 = z_1 + h z'_1 = -0.075 + 0.5[0.05(-0.075) - 0.15(1)] = -0.15187$$

Por tanto, la solución buscada es

$$y(1) = y_2 = 0.96250$$

$$y'(1) = z(1) = z_2 = -0.15187$$

EJEMPLO 7.10

Resolver por el método de Euler $y' = y - t^2 + 1$, con la condición inicial $y(0) = y_0 = 0.5$, en el intervalo $0 \le t \le 2$, con un paso de $h = 0.2$.

El esquema numérico para esta ecuación queda de la siguiente manera:

$$y_n = y_{n-1} + h\left[y_{n-1} - t_{n-1}^2 + 1\right]$$

La condición inicial es $t_0 = 0 \quad y_0 = 0.5$. Así, los demás pasos hasta completar el intervalo son

$$t_1 = 0.2 \quad y_1 = 0.5 + 0.2\left[0.5 + (0)^2 + 1\right] = 0.8000$$

$$t_2 = 0.4 \quad y_2 = 0.8 + 0.2\left[0.8 - (0.2)^2 + 1\right] = 1.1520$$

$$t_3 = 0.6 \quad y_3 = 1.1520 + 0.2\left[1.1520 - (0.4)^2 + 1\right] = 1.5504$$

$$t_4 = 0.8 \quad y_4 = 1.5504 + 0.2\left[1.5504 - (0.6)^2 + 1\right] = 1.9884$$

$$t_5 = 1.0 \quad y_5 = 1.9884 + 0.2\left[1.9884 - (0.8)^2 + 1\right] = 2.4581$$

www.FreeLibros.me

$$t_6 = 1.2 \quad y_6 = 2.4581 + 0.2\left[2.4581 - (1.0)^2 + 1\right] = 2.9498$$

$$t_7 = 1.4 \quad y_7 = 2.9498 + 0.2\left[2.9498 - (1.2)^2 + 1\right] = 3.4517$$

$$t_8 = 1.6 \quad y_8 = 3.4517 + 0.2\left[3.4517 - (1.4)^2 + 1\right] = 3.9501$$

$$t_9 = 1.8 \quad y_9 = 3.9501 + 0.2\left[3.9501 - (1.6)^2 + 1\right] = 4.4281$$

$$t_{10} = 2.0 \quad y_{10} = 4.8657 + 0.2\left[4.4281 - (1.8)^2 + 1\right] = 4.8657$$

7.2.3 Métodos Runge-Kutta

El método de la serie de Taylor de la sección anterior presenta la desventaja de necesitar análisis previo a su programación. Por ello, si se desea utilizar el método de la serie de Taylor de cuarto orden en el problema general

$$\begin{aligned} y' &= f(t, y) \\ y(t_0) &= y_0 \end{aligned} \tag{7.11}$$

se debe derivar sucesivamente la ecuación diferencial para determinar las fórmulas para $y'', y''', y^{(4)}$. Una vez hecho esto, las funciones se deberán programar. Los métodos de Runge-Kutta evitan esta dificultad [Nakamura, 1992], [Maron *et al.*, 1995], [Mathews *et al.*, 2000], [Burden *et al.*, 2002], [Nieves *et al.*, 2002], [Rodríguez, 2003], a pesar de que imitan a los métodos de la serie de Taylor mediante una combinación ingeniosa de los valores de $f(t, y)$. Así, considerando un método de la forma

$$y_{n+1} = y_n + h\sum_{i=1}^{r} w_i k_i = y_n + h(w_1 k_1 + w_2 k_2 + \cdots + w_r k_r) \tag{7.12}$$

donde $k_1 = f(t_n, y_n)$ y $k_i = f(t_n + h\alpha_i, y_n + h\beta_i k_{i-1})$, $i > 1$, determinando los coeficientes α_i, β_i, y w_i de forma que los primeros $p+1$ términos de la expansión de la serie de Taylor de (7.7) coincidan con los primeros $p+1$ términos de (7.12). Para esto se considera la serie de Taylor en dos variables; es decir, se tiene que

$$\begin{aligned} & f(t_n + h\alpha, y_n + h\beta k) = \\ & f(t_n, y_n) + h\left(\alpha\frac{\partial}{\partial t} + \beta k\frac{\partial}{\partial y}\right) f(t_n, y_n) + \frac{h^2}{2!}\left(\alpha\frac{\partial}{\partial t} + \beta k\frac{\partial}{\partial y}\right)^2 f(t_n, y_n) \\ & + \cdots + \frac{h^{p-1}}{(p-1)!}\left(\alpha\frac{\partial}{\partial t} + \beta k\frac{\partial}{\partial y}\right)^{p-1} f(t_n, y_n) + h^p S_p(t_n, y_n, \alpha, \beta k) \end{aligned} \tag{7.13}$$

donde $h^p S_p(t_n, y_n, \alpha, \beta k)$ es el residuo. Está claro que para igualar los términos de (7.12) usando la serie de Taylor (7.13), con los términos de la expansión de (7.7) usar (7.12) es demasiado tedioso. A continuación se ilustra el caso para $p = 2$.

7.2.3.1 Métodos de Runge-Kutta de segundo orden

La serie de Taylor dada por (7.7) es

$$y_{n+1} = y_n + hf^{(0)} + \frac{h^2}{2!} f^{(1)} + \frac{h^3}{3!} f^{(2)} + \cdots + \frac{h^p}{p!} f^{(p-1)}$$

donde

$$f^{(0)} = f$$

www.FreeLibros.me

$$f^{(1)} = f_t + f_y f$$

$$f^{(2)} = f_{tt} + 2f_{ty} f + f_{yy} f^2 + (f_t + f_y f) f_y$$

Considerando la ecuación (7.12) con $p = 2$, se tiene

$$y(t_{n+1}) = y(t_n) + hf + \frac{h^2}{2}(f_t + f_y f) + h^3 E_3(t_n), \tag{7.14}$$

donde f, f_t y f_y están evaluadas en $(t_n, y(t_n))$. Claramente se tiene una infinidad de formas de elegir los valores de r, w_i, α_i y β_i. Si se toma $r = 2$ en (7.12), al expandirla se transforma en

$$y_{n+1} = y_n + h(w_1 k_1 + w_2 k_2)$$

donde

$$k_1 = f(t_n, y_n) = f$$

$$k_2 = f(t_n + h\alpha_2, y_n + h\beta_2 k_1)$$

De la ecuación (7.13), se tiene que

$$k_2 = f(t_n + h\alpha_2, y_n + h\beta_2 k_1) = f(t_n, y_n) + h\left(\alpha_2 \frac{\partial}{\partial t} + \beta_2 k_1 \frac{\partial}{\partial y}\right) f(t_n, y_n)$$

Simplificando términos se obtiene

$$k_2 = f + h\alpha_2 f_t + h\beta_2 k_1 f_y$$

Sustituyendo los valores de $k_1 = f$ y $k_2 = f + h\alpha_2 f_t + h\beta_2 k_1 f_y$, se llega a

$$y_{n+1} = y_n + h(w_1 f + w_2 (f + h\alpha_2 f_t + h\beta_2 k_1 f_y))$$

Sustituyendo nuevamente $k_1 = f$ y realizando las operaciones algebraicas, se obtiene

$$y_{n+1} = y_n + hw_1 f + hw_2 f + h^2 w_2 \alpha_2 f_t + h^2 w_2 \beta_2 f f_y$$

Reagrupando términos, finalmente se llega a

$$y_{n+1} = y_n + h(w_1 + w_2) f + h^2 w_2 (\alpha_2 f_t + \beta_2 f f_y) \tag{7.15}$$

donde los argumentos de f, f_t y f_y son (t_n, y_n). Al comparar (7.14) con (7.15) se observa que es necesario que

$$w_1 + w_2 = 1 \qquad w_2\alpha_2 = \frac{1}{2} \qquad \text{y} \qquad w_2\beta_2 = \frac{1}{2}$$

7.2.3.1.1 Método del punto medio

Una solución al sistema de ecuaciones anterior es: $w_1 = 0$, $w_2 = 1$ y $\alpha_2 = \beta_2 = \frac{1}{2}$, y con esta elección se tiene que la ecuación (7.12) se transforma en

$$y_{n+1} = y_n + hf\left(t_n + \frac{1}{2}h,\ y_n + \frac{1}{2}hf(t_n, y_n)\right),$$

lo cual se llama *método del punto medio*. Si se hace que $k_1 = hf(t_n, y_n)$, se obtiene

$$y_{n+1} = y_n + hf\left(t_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1\right)$$

www.FreeLibros.me

7.2.3.1.2 Método de Euler modificado

Otra elección de los parámetros es: $w_1 = w_2 = \frac{1}{2}$ y $\alpha_2 = \beta_2 = 1$, con la cual se obtiene

$$y_{n+1} = y_n + \frac{1}{2}h[f(t_n, y_n) + f(t_n + h, y_n + hf(t_n, y_n))],$$

conocido como el *método de Euler modificado*.

Si se hace que $t_n + h = t_{n+1}$, $k_1 = hf(t_n, y_n)$, se obtiene

$$y_{n+1} = y_n + \frac{1}{2}[k_1 + hf(t_{n+1}, y_n + k_1)]$$

Si adicionalmente se hace que $k_2 = hf(t_{n+1}, y_n + k_1)$, se llega finalmente a la expresión

$$y_{n+1} = y_n + \frac{1}{2}[k_1 + k_2]$$

La sección 7.9.3 proporciona el código computacional de este método desarrollado en Matlab.

EJEMPLO 7.11

Con el método de Runge-Kutta, denominado Euler modificado, resolver la ecuación $i' = 0.2 - 0.4i$, en el intervalo en segundos, $0 < t \le 1$, con un paso de $h = 0.1$, sujeta a la condición inicial $i(0) = 0$.

La función de la derivada se expresa como

$$f(i_{n+1}, t_{n+1}) = 0.2 - 0.4i_n$$

De aquí se obtiene que

$$f(i_n + k_1, t_{n+1}) = 0.2 - 0.4(i_n + k_1)$$

$$k_1 = h\, f(i_n, t_{n+1}) = h[0.2 - 0.4i_n]$$

$$k_2 = h\, f(i_n + k_1, t_{n+1}) = h[0.2 - 0.4(i_n + k_1)]$$

Así, el esquema numérico final es

$$i_{n+1} = i_n + \frac{1}{2}(k_1 + k_2)$$

Si se tiene la condición inicial dada por

$$t_0 = 0,\ i_0 = 0$$

El siguiente valor es

$$t_1 = 0.1$$

$$k_1 = h(0.2 - 0.4i_0) = 0.1(0.2 - 0.4(0)) = 0.02$$

$$k_2 = h[0.2 - 0.4(i_0 + k_1)] = 0.1[0.2 - 0.4(0 + 0.02)] = 0.0192$$

$$i_1 = i_0 + \frac{1}{2}(k_1 + k_2) = 0 + \frac{1}{2}(0.02 + 0.0198) = 0.0196$$

La siguiente tabla resume los resultados numéricos:

www.FreeLibros.me

Tabla 7.1 Resultados de aplicar el método de Runge-Kutta, denominado de Euler modificado a la expresión $i' = 0.2 - 0.4i$.

$t_0 = 0$			$i_0 = 0$
$t_1 = 0.1$	$k_1 = 0.0200$	$k_2 = 0.0192$	$i_1 = 0.0196$
$t_2 = 0.2$	$k_1 = 0.0192$	$k_2 = 0.0184$	$i_2 = 0.0384$
$t_3 = 0.3$	$k_1 = 0.0185$	$k_2 = 0.0177$	$i_3 = 0.0565$
$t_4 = 0.4$	$k_1 = 0.0177$	$k_2 = 0.0170$	$i_4 = 0.0739$
$t_5 = 0.5$	$k_1 = 0.0170$	$k_2 = 0.0164$	$i_5 = 0.0906$
$t_6 = 0.6$	$k_1 = 0.0164$	$k_2 = 0.0157$	$i_6 = 0.1067$
$t_7 = 0.7$	$k_1 = 0.0157$	$k_2 = 0.0151$	$i_7 = 0.1221$
$t_8 = 0.8$	$k_1 = 0.0151$	$k_2 = 0.0145$	$i_8 = 0.1369$
$t_9 = 0.9$	$k_1 = 0.0145$	$k_2 = 0.0139$	$i_9 = 0.1511$
$t_{10} = 1.0$	$k_1 = 0.0140$	$k_2 = 0.0134$	$i_{10} = 0.1648$

La solución analítica de esta ecuación es $i(t) = \frac{1}{2} - \frac{1}{2}e^{-0.4t}$. La figura 7.2 muestra la comparación entre la solución numérica y la solución analítica. Se puede observar que, a pesar de que el paso de discretización es bastante grande, el método numérico es bastante preciso.

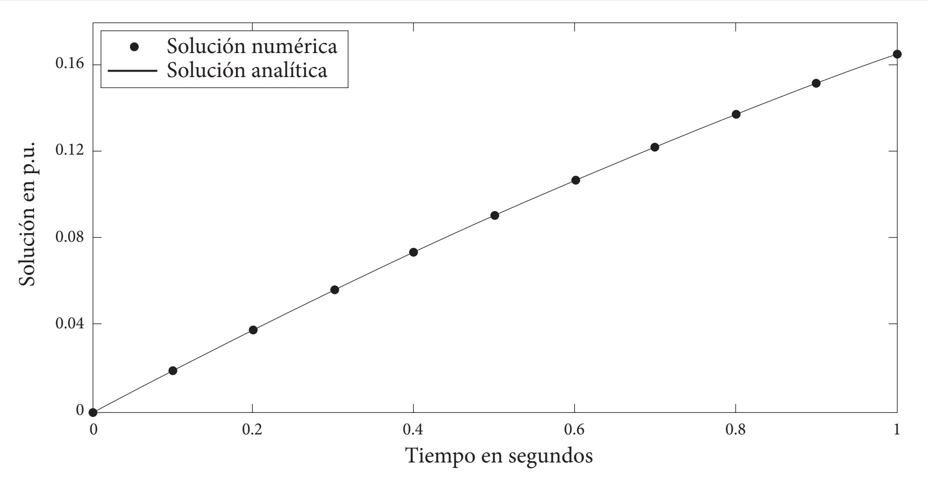


Figura 7.2 Solución numérica *vs.* solución analítica.

7.2.3.1.3 Método de Heun

Si se elige $w_1 = \frac{1}{4}$, $w_2 = \frac{3}{4}$ y $\alpha_2 = \beta_2 = \frac{2}{3}$, se obtiene

$$y_{n+1} = y_n + \frac{1}{4}h\left[f(t_n, y_n) + 3f\left(t_n + \frac{2}{3}h, y_n + \frac{2}{3}hf(t_n, y_n)\right)\right]$$

lo cual se llama *método de Heun*.

Si se define $k_1 = hf(t_n, y_n)$, se obtiene

$$y_{n+1} = y_n + \frac{1}{4}\left[k_1 + 3hf\left(t_n + \frac{2}{3}h, y_n + \frac{2}{3}k_1\right)\right]$$

www.FreeLibros.me

Si además se define $k_2 = hf\left(t_n + \frac{2}{3}h,\ y_n + \frac{2}{3}k_1\right)$, se llega finalmente a la expresión

$$y_{n+1} = y_n + \frac{1}{4}[k_1 + 3k_2]$$

7.2.3.2 Métodos de Runge-Kutta de tercer orden

El método de Runge-Kutta de tercer orden se obtiene al tomar $p = 4$ y $r = 3$. Este método no se deduce aquí, pero se presenta una de las fórmulas que lo definen, la cual tiene la siguiente estructura

$$y_{n+1} = y_n + \frac{h}{9}[2k_1 + 3k_2 + 4k_3]$$

donde,

$$k_1 = f(t_n, y_n)$$
$$k_2 = f\left(t_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_1\right)$$
$$k_3 = f\left(t_n + \frac{3}{4}h, y_n + \frac{3}{4}hk_2\right)$$

Otra posible elección de los parámetros lleva a otro método de Runge-Kutta de tercer orden de la forma

$$y_{n+1} = y_n + \frac{h}{6}[k_1 + 4k_2 + k_3]$$

donde

$$k_1 = f(t_n, y_n)$$
$$k_2 = f\left(t_n + \frac{h}{2}, y_n + \frac{1}{2}hk_1\right)$$
$$k_3 = f(t_n + h, y_n + hk_1)$$

La sección 7.9.4 proporciona el código desarrollado en Matlab del método de Runge-Kutta de tercer orden. En este caso específico se desarrolla la segunda formulación obtenida.

7.2.3.3 Método de Runge-Kutta clásico

El método de Runge-Kutta clásico o de cuarto orden se obtiene al tomar $p = 4$ y $r = 4$ y está definido por

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$

donde $k_1 = f(t_n, y_n)$, $k_2 = f\left(t_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_1\right)$, $k_3 = f\left(t_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_2\right)$ y $k_4 = f(t_n + h, y_n + hk_3)$.

Existen métodos del tipo Runge-Kutta de mayor orden. Éstos se emplearán en secciones posteriores. La sección 7.9.5 proporciona el código desarrollado en Matlab donde se implementa numéricamente el método de Runge-Kutta de cuarto orden.

EJEMPLO 7.12

Utilizando el método de Runge-Kutta clásico, para la ecuación $y' = -2y$, con la condición inicial $y(0) = 1$ y un paso de $h = 0.1$; obtener los valores en el intervalo $0 \le x \le 0.2$.

www.FreeLibros.me

La condición inicial indica que

$$x = 0 \qquad y_0 = 1$$

El siguiente valor es

$$x = 0.1$$

$$k_1 = h\,f(x_0, y_0) = 0.1[-2(1)] = -0.2$$

$$k_2 = h\,f\left(x_0 + \frac{h}{2}, y_0 + \frac{k_1}{2}\right) = 0.1\left[-2\left(1 - \frac{0.2}{2}\right)\right] = -0.18$$

$$k_3 = h\,f\left(x_0 + \frac{h}{2}, y_0 + \frac{k_2}{2}\right) = 0.1\left[-2\left(1 - \frac{0.18}{2}\right)\right] = -0.182$$

$$k_4 = h\,f(x_0 + h, y_0 + k_3) = 0.1[-2(1 - 0.182)] = -0.1636$$

$$\begin{aligned} y_1 &= y_0 + \frac{1}{6}[k_1 + 2k_2 + 2k_3 + k_4] \\ &= 1 + \frac{1}{6}[(-0.2) + 2(-0.18) + 2(-0.182) + (-0.1636)] \\ &= 0.8187 \end{aligned}$$

El último valor buscado es, por tanto,

$$x = 0.2$$

$$k_1 = h\,f(x_1, y_1) = 0.1[-2(0.8187)] = -0.1637$$

$$k_2 = h\,f\left(x_1 + \frac{h}{2}, y_1 + \frac{k_1}{2}\right) = 0.1\left[-2\left(0.8187 - \frac{0.1627}{2}\right)\right] = -0.1474$$

$$k_3 = h\,f\left(x_1 + \frac{h}{2}, y_1 + \frac{k_2}{2}\right) = 0.1\left[-2\left(0.8187 - \frac{0.1474}{2}\right)\right] = -0.1490$$

$$k_4 = h\,f(x_1 + h, y_1 + k_3) = 0.1[-2(0.8187 - 0.1490)] = -0.1339$$

$$\begin{aligned} y_2 &= y_1 + \frac{1}{6}[k_1 + 2k_2 + 2k_3 + k_4] \\ &= 0.8187 + \frac{1}{6}[(-0.1637) + 2(-0.1474) + 2(-0.1490) + (-0.1339)] \\ &= 0.6705 \end{aligned}$$

7.3 Consistencia, convergencia y estabilidad de los métodos de un paso

Dado un esquema numérico para resolver una ecuación diferencial, existen criterios para evaluar su efectividad, estos criterios están relacionados con el grado de precisión con que se aproxima la solución dada por el esquema numérico a la solución de la ecuación diferencial.

www.FreeLibros.me

7.3.1 Consistencia

La forma general de los métodos de un paso para resolver

$$y' = f(t, y),\ y(t_0) = y_0 \tag{7.16}$$

es

$$y_{n+1} = y_n + h\phi(t_n, y_n; h) \tag{7.17}$$

Si $h \neq 0$, se puede escribir en forma genérica como

$$\frac{y_{n+1} - y_n}{h} = h\phi(t_n, y_n; h) \tag{7.18}$$

El interés se centra en la precisión de (7.17) para aproximar a (7.16). Si $y(t)$ es la solución de (7.16), se define el *error de truncamiento* de (7.17) como

$$\tau(t;h) = \frac{y(t+h) - y(t)}{h} - h\phi(t, y(t); h) \tag{7.19}$$

donde t está en el intervalo $[t_0, b]$ y $h > 0$. Se dice que (7.17) es un método consistente con (7.4), o simplemente consistente, si

$$\tau(t;h) \to 0 \text{ cuando } h \to 0$$

de manera uniforme para t en el intervalo $[t_0, b]$. Se puede observar de (7.18) que, para que un método sea consistente, se debe cumplir que $\phi(t, y(t);0) = f(t, y(t))$.

Se dice que el método es consistente de orden p si existe $M \geq 0$, $h_0 > 0$ y un entero positivo p tal que

$$\sup_{t_0 \leq t \leq b} |\tau(t;h)| \leq Mh^p, \text{ para toda } h \text{ en } (0, h_0] \tag{7.20}$$

El método de la serie de Taylor es consistente de orden p ya que

$$\tau(t;h) = h^{p+1} E_{p+1}(t) = \frac{h^{p+1}}{(p+1)!} y^{(p+1)}(\xi)$$

donde $t < \xi < t + h$. Por lo que se elige

$$M = \frac{1}{(p+1)!} \max_{t_0 \leq t \leq b} \left| y^{(p+1)}(t) \right|$$

Los métodos Runge-Kutta con $p+1$ términos que coinciden con la expansión de la serie de Taylor también son consistentes de orden p. Se puede afirmar intuitivamente de (7.18) que si un método es consistente, entonces el esquema numérico (7.17) es una buena aproximación a la ecuación diferencial (7.16).

7.3.2 Convergencia

Hasta el momento se ha considerado sólo el error de truncamiento. Éste mide la precisión con que la ecuación en diferencias (7.17) se aproxima a la ecuación diferencial. No se ha considerado con qué precisión la solución de (7.17) se aproxima a la solución exacta $y(t)$ de la ecuación diferencial original. Además, lo que interesa es saber si la solución de la ecuación en diferencias converge a la solución de la ecuación diferencial conforme el tamaño de paso h se reduce.

Se define la convergencia en forma cuidadosa. Si se observa el comportamiento de y_n conforme $h \to 0$ y n se mantiene fijo, no se obtiene un concepto útil. Resulta claro que se tiene

$$t_n = t_0 + nh \to t_0 \text{ cuando } h \to 0, \text{ con } n \text{ fijo}$$

www.FreeLibros.me

Por tanto, se debe considerar el comportamiento de y_n cuando $h \to 0$ con $t_n = t_0 + nh$ fijo. A fin de obtener una solución en un valor fijo $t_n \neq t_0$, se debe incrementar el número n de pasos conforme $h \to 0$. Si

$$\lim_{\substack{n \to 0 \\ t_n = t \text{ fijo}}} y_n = y(t)$$

para toda $t \in [t_0, b]$, se dice que el método es convergente.

El siguiente teorema muestra una cota para el error de un método consistente de un paso, con tal de que la función ϕ que define al método satisfaga una condición de Lipschitz con respecto a y. De esto se infiere que este tipo de métodos son convergentes.

Teorema 7.3

Considerando que el problema de valor inicial

$$y' = f(t, y), \quad y(t_0) = y_0, \quad t \in [t_0, b]$$

se reemplaza por un método de un paso como

$$\begin{aligned} y_0 &= y_0 \\ t_n &= t_0 + nh \\ y_{n+1} &= y_n + h\phi(t_n, y_n; h) \end{aligned}$$

donde ϕ satisface la condición de Lipschitz $|\phi(t, y; h) - \phi(t, z; h)| \leq L_\phi |y - z|$, para toda $t \in [t_0, b]$, $-\infty < y$, $z < \infty$, $h \in [0, h_0]$ para algún L_ϕ y $h_0 > 0$.

Si el método es consistente de orden p, de modo que el error de truncamiento definido por (7.19) satisface (7.20), entonces el error global $y(t_n) - y_n$ satisface

$$|y(t_n) - y_n| \leq \begin{cases} \left(\frac{e^{(t_n - t_0)L_\phi - 1}}{L_\phi}\right) Mh^p & L_\phi \neq 0 \\ (t_n - t_0) Mh^p & L_\phi = 0 \end{cases} \tag{7.21}$$

para toda $t_n \in [t_0, b]$ y toda $h \in (0, h_0)]$ y, por tanto, el método es convergente. El cálculo de L_ϕ es relativamente sencillo como lo muestran los siguientes ejemplos.

EJEMPLO 7.13

Para el método de Euler se tiene que $\phi(t, y; h) = f(t, y)$ por lo que se deduce que

$$|\phi(t, y; h) - \phi(t, z; h)| = |f(t, y) - f(t, z)| \leq L|y - z|$$

donde L es la constante de Lipschitz para f.

EJEMPLO 7.14

Para el método de Taylor de orden p se define

$$L_k = \max_{\substack{t \in [a, b] \\ -\infty < y < \infty}} \left| \frac{\partial f^{(k)}(t, y)}{\partial y} \right|$$

Estas constantes existen, ya que se supone que f es lo suficientemente suave. Para el método de Taylor de orden p se tiene

$$\phi(t, y; h) = f(t, y) + \frac{h}{2!} f^{(1)}(t, y) + \cdots + \frac{h^{p-1}}{p!} f^{p-1}(t, y)$$

www.FreeLibros.me

por tanto,

$$|\phi(t,y;h)-\phi(t,z;h)|=\left|\left(f(t,y)+\frac{h}{2!}f^{(1)}(t,y)+\cdots+\frac{h^{p-1}}{p!}f^{p-1}(t,y)\right)-\left(f(t,z)+\frac{h}{2!}f^{(1)}(t,z)+\cdots+\frac{h^{p-1}}{p!}f^{p-1}(t,z)\right)\right|$$

$$\leq|f(t,y)-f(t,z)|+\frac{h}{2!}|f^{(1)}(t,y)-f^{(1)}(t,z)|+\cdots+\frac{h^{p-1}}{p!}|f^{(p-1)}(t,y)-f^{(p-1)}(t,z)|$$

Usando el teorema del valor medio para cada $f^{(k)}$ se obtiene

$$|\phi(t,y;h)-\phi(t,z;h)|\leq L_0|y-z|+\frac{h}{2!}L_1|y-z|+\cdots+\frac{h^{p-1}}{p!}L_{p-1}|y-z|$$

$$=\left(L_0+\frac{h}{2!}L_1+\cdots+\frac{h^{p-1}}{p!}L_{p-1}\right)|y-z|$$

De donde

$$L_\phi=\left(L_0+\frac{h}{2!}L_1+\cdots+\frac{h^{p-1}}{p!}L_{p-1}\right)$$

EJEMPLO 7.15

Para el método del punto medio

$$y_{n+1}=y_n+hf\left(t_n+\frac{1}{2}h,\,y_n+\frac{1}{2}hf(t_n,y_n)\right)$$

se tiene que

$$\phi(t,y;h)=f\left(t+\frac{1}{2}h,\,y+\frac{1}{2}hf(t,y)\right)$$

por lo que

$$|\phi(t,y;h)-\phi(t,z;h)|=\left|f\left(t+\frac{1}{2}h,\,y+\frac{1}{2}hf(t,y)\right)-f\left(t+\frac{1}{2}h,\,z+\frac{1}{2}hf(t,z)\right)\right|$$

$$|\phi(t,y;h)-\phi(t,z;h)|\leq L\left|\left(y+\frac{1}{2}hf(t,y)\right)-\left(z+\frac{1}{2}hf(t,z)\right)\right|$$

$$|\phi(t,y;h)-\phi(t,z;h)|\leq L|y-z|+\frac{1}{2}hL|f(t,y)-f(t,z)|$$

$$|\phi(t,y;h)-\phi(t,z;h)|\leq L|y-z|+\frac{1}{2}hL^2|y-z|$$

$$|\phi(t,y;h)-\phi(t,z;h)|=\left(L+\frac{1}{2}hL^2\right)|y-z|$$

donde L es la constante de Lipschitz para f. De esto se infiere

$$L_\phi=L+\frac{1}{2}hL^2$$

www.FreeLibros.me

7.3.3 Estabilidad

Un buen método numérico de un paso debe garantizar que pequeñas perturbaciones, inducidas tal vez por errores de redondeo en el cálculo de la aproximación produzcan cambios razonablemente pequeños en las aproximaciones posteriores. Esta propiedad se llama *estabilidad del método*. A continuación se da la definición formal.

Definición 7.2 Un esquema numérico de un paso

$$\begin{aligned} y_0 &= y_0 \\ t_n &= t_0 + nh \\ y_{n+1} &= y_n + h\phi(t_n, y_n; h) \end{aligned}$$

para resolver el problema de valor inicial

$$y' = f(t, y),\ y(t_0) = y_0,\ t \in [t_0, b]$$

se dice que es estable si existen constantes positivas K, h_0, independientes de n, tales que si $0 < h \leq h_0$, $|v_n - y_n| \leq K|v_0 - y_0|$, $n = 0, 1, 2, \ldots, N$, donde v_0 es otro valor inicial para la ecuación diferencial. Así, se tiene que un método de un paso como el anterior es estable si $\phi(t, y; h)$ satisface una condición de Lipschitz L_ϕ en la segunda variable. Se puede observar que esta condición se usa para demostrar que un método es convergente.

7.3.4 Error de redondeo y métodos de un paso

Considerando el efecto del error de redondeo al calcular la solución del método de un paso

$$y_{n+1} = y_n + h\phi(t_n, y_n; h)$$

Dada una precisión aritmética, se demostrará que existe un tamaño de paso h_1 tal que si $h < h_1$ entonces la influencia del error de redondeo se amplificará. Para esto, se considera que en lugar de calcular la solución de (7.17) se calcula la solución de

$$w_{n+1} = w_n + h\phi(t_n, w_n; h) + \varepsilon_n$$

donde ε_n es el error debido al redondeo. Se puede demostrar que (7.20) se transforma en

$$|y(t_n) - w_n| \leq \begin{cases} \left(\frac{e^{(t_n - t_0)L_\phi} - 1}{L_\phi}\right)\left(Mh^p + \dfrac{\varepsilon}{h}\right), & L_\phi \neq 0 \\ (t_n - t_0)\left(Mh^p + \dfrac{\varepsilon}{h}\right), & L_\phi = 0 \end{cases}$$

donde $\varepsilon = \sup_{n=0,1,\ldots} |\varepsilon_n|$. De esto se tiene que el error global se puede incrementar al reducir el tamaño de paso h.

7.3.5 Control del error

Con una adecuada elección del tamaño de paso h es posible reducir el error de truncamiento tanto como se desee, suponiendo que se está efectuando una aritmética exacta. Para esto se necesitan considerar dos métodos de un paso con diferentes órdenes. Suponiendo que se tienen dos métodos, uno de orden p y otro de orden $p+1$. Sean

$$y_{n+1} = y_n + h\phi(t_n, y_n; h) \tag{7.22a}$$

www.FreeLibros.me

$$w_{n+1} = w_n + h\psi(t_n, w_n; h) \tag{7.22b}$$

los métodos. Entonces, suponiendo que $y_n \approx y(t_n) \approx w_n$, se puede demostrar que

$$\tau_{i+1} = \tau(t_{i+1}; h) \approx \frac{1}{h}(y_{i+1} - w_{i+1})$$

donde $\tau(t_{i+1}; h)$ es el error de truncamiento del método de orden p. Esta última forma es una estimación del error local de truncamiento. El objetivo no sólo es estimar este error, sino lograr que éste sea menor que una cota dada. Para este fin, dado que (7.22a) es un método de orden p, se infiere que existe $K > 0$ tal que $\tau_{i+1} = \tau_{i+1}(h) \approx Kh^p$.

Suponiendo que se reduce el tamaño de paso por un factor r, esto es, considerando el nuevo tamaño de paso como rh, se tiene que

$$\tau_{i+1}(rh) \approx K(rh)^p = r^p(Kh^p) \approx r^p \tau_{i+1}(h) \approx \frac{r^p}{h}(y_{i+1} - w_{i+1})$$

Si se espera que el error de truncamiento esté acotado por $\varepsilon > 0$, se debe elegir r de modo que

$$\frac{r^p}{h}|y_{i+1} - w_{i+1}| \approx |\tau_{i+1}(rh)| \le \varepsilon$$

Despejando r se obtiene

$$r \le \left(\frac{\varepsilon h}{|y_{i+1} - w_{i+1}|}\right)^{\frac{1}{p}}$$

El método más conocido que usa este criterio para cambiar el tamaño de paso para controlar el error de truncamiento es el método de Runge-Kutta-Fehlberg. Esta idea combina dos métodos del tipo Runge-Kutta, uno de orden 4 y otro de orden 5. La idea genial detrás de este método es que los k_i's que intervienen en cada uno de los métodos coinciden. Este método está formado por el método de orden 4

$$y_{n+1} = y_n + h\left(\frac{25}{216}k_1 + \frac{1408}{2565}k_3 + \frac{2197}{4104}k_4 - \frac{1}{5}k_5\right)$$

y el método de orden 5

$$w_{n+1} = y_n + h\left(\frac{16}{135}k_1 + \frac{6656}{12825}k_3 + \frac{28561}{56430}k_4 - \frac{9}{50}k_5 + \frac{2}{55}k_6\right)$$

donde

$$k_1 = f(t_n, y_n)$$

$$k_2 = f\left(t_n + \frac{1}{4}h, y_n + \frac{1}{4}hk_1\right)$$

$$k_3 = f\left(t_n + \frac{3}{8}h, y_n + \frac{3}{32}hk_1 + \frac{9}{32}hk_2\right)$$

$$k_4 = f\left(t_n + \frac{12}{13}h, y_n + \frac{1932}{2197}hk_1 - \frac{7200}{2197}hk_2 + \frac{7296}{2197}hk_3\right)$$

$$k_5 = f\left(t_n + h, y_n + \frac{439}{216}hk_1 - 8hk_2 + \frac{3680}{513}hk_3 - \frac{845}{4104}hk_4\right)$$

$$k_6 = f\left(t_n + \frac{1}{2}h, y_n - \frac{8}{27}hk_1 + 2hk_2 - \frac{3544}{2656}hk_3 + \frac{1859}{4104}hk_4 - \frac{11}{40}hk_5\right)$$

www.FreeLibros.me

Cabe hacer notar que el cálculo de w_{n+1} no es necesario ya que sólo se requiere el de $|y_{i+1}-w_{i+1}|$ y éste está dado por

$$|y_{i+1}-w_{i+1}|=h\left|\frac{1}{360}k_1-\frac{128}{4275}k_3-\frac{2197}{75240}k_4+\frac{1}{50}k_5+\frac{2}{55}k_6\right|$$

El valor de r así calculado tiene los objetivos de rechazar la elección de h si el error de truncamiento no cumple el criterio establecido y predecir una elección adecuada de h para el paso siguiente.

Para la implementación práctica del método se toma

$$r=\left(\frac{\varepsilon h}{2|y_{i+1}-w_{i+1}|}\right)^{\frac{1}{p}}$$

Además, se debe recordar no reducir el tamaño de h si el error satisface las especificaciones, y no aumentar demasiado el tamaño de paso cuando éste sea demasiado grande. Existe otro método basado en las mismas ideas; éste es el método de Runge-Kutta-Merson. También se tiene una estimación del error de truncamiento que se puede usar para estimar el tamaño de paso adecuado como en el caso del método de Runge-Kutta-Fehlberg. Las fórmulas que lo definen son

$$k_1=f(t_n,y_n) \qquad k_2=f\left(t_n+\frac{h}{3},y_n+\frac{h}{3}k_1\right)$$

$$k_3=f\left(t_n+\frac{h}{3},y_n+\frac{h}{6}k_1+\frac{h}{6}k_2\right) \qquad k_4=f\left(t_n+\frac{h}{2},y_n+\frac{h}{8}k_1+\frac{3h}{8}k_3\right)$$

$$k_5=f\left(t_n+h,y_n+\frac{h}{2}k_1-\frac{3h}{2}k_3+2hk_4\right)$$

El esquema numérico es, entonces

$$y_{n+1}=y_n+\frac{h}{6}(k_1+4k_4+k_5)$$

Este método tiene un error de truncamiento de orden 4. Siguiendo la notación usada en el método de Runge-Kutta-Fehlberg, se tiene que

$$|y_{i+1}-w_{i+1}|=\frac{h}{30}|(2k_1-9k_3+8k_4-k_5)|$$

La implementación del método es idéntica al anterior; sólo hay que considerar los cambios correspondientes a cada método. Los métodos de este tipo, donde el tamaño de paso se cambia para satisfacer algún criterio de reducción del error, reciben el nombre de *métodos adaptativos*.

7.4 Métodos multipaso basados en integración numérica

Dada la ecuación diferencial $y'=f(t,y)$, integrando desde t_n hasta t_{n+1} y aplicando el teorema fundamental del cálculo, se obtiene

$$y(t_{n+1})-y(t_n)=\int_{t_n}^{t_{n+1}}f(t,y(t))dt \tag{7.23}$$

La base de muchos métodos numéricos para resolver ecuaciones diferenciales es reemplazar la integral del lado derecho en (7.23) por una aproximación adecuada. Por ejemplo, se puede reemplazar $f(t,y(t))$ por su polinomio de Taylor alrededor de $t=t_n$ e integrar este polinomio. Este procedimiento conduce al método de la serie de Taylor. Un procedimiento mejor es reemplazar f por un polinomio interpolador. Suponiendo que se conocen las aproximaciones $y_0, y_1, \ldots, y_n$ de $y(t)$ en los puntos $t_k=t_0+kh$, $k=0,1,\ldots,n$; se tiene que

www.FreeLibros.me

$$f_k = f(t_k, y_k)$$

7.4.1 Métodos explícitos

El polinomio interpolador en los $m+1$ puntos (t_{n-m}, f_{n-m}), $(t_{n-m+1}, f_{n-m+1}),\ldots,(t_{n-1}, f_{n-1})$, (t_n, f_n) con $m \le n$ se usa como una aproximación a $f(t, y(t))$ en el intervalo (t_n, t_{n+1}). De (7.23) se obtiene [Burden *et al.*, 2002]

$$y(t_{n+1}) - y(t_n) = \int_{t_n}^{t_{n+1}} f(t, y(t))dt$$

Sustituyendo el integrando por medio de la expresión

$$f(x) = \sum_{k=0}^{n} \nabla^k y_n (-1)^k \binom{-s}{k} + h^{n+1}(-1)^{n+1}\binom{-s}{n+1} f^{(n+1)}(\xi), \quad x = x_n + sh$$

y efectuando el cambio de variable $t = t_n + sh$, se tiene

$$y(t_{n+1}) - y(t_n) = h\int_0^1 \left(\sum_{k=0}^{m} \nabla^k f_n (-1)^k \binom{-s}{k} + h^{m+1}(-1)^{m+1}\binom{-s}{m+1} f^{(m+1)}(\xi)\right) ds$$

Separando las integrales se deduce que

$$\begin{aligned} y(t_{n+1}) - y(t_n) &= h\sum_{k=0}^{m} \nabla^k f_n \left(\int_0^1 (-1)^k \binom{-s}{k} ds \right) + h^{m+2}\int_0^1 (-1)^{m+1}\binom{-s}{m+1} f^{(m+1)}(\xi) ds \\ &= h\sum_{k=0}^{m} \nabla^k f_n \left(\int_0^1 (-1)^k \binom{-s}{k} ds \right) + h^{m+2} f^{(m+1)}(\xi_n)\int_0^1 (-1)^{m+1}\binom{-s}{m+1} ds \end{aligned}$$

donde se usó el teorema del valor medio en la evaluación de la integral.

Definiendo

$$b_k = \left(\int_0^1 (-1)^k \binom{-s}{k} ds \right)$$

se obtiene

$$y(t_{n+1}) = y(t_n) + h\sum_{k=0}^{m} b_k \nabla^k f_n + h^{m+2} b_{m+1} f^{(m+1)}(\xi_n) \tag{7.24}$$

A partir de esta expresión, si se ignora el último término, se obtiene el algoritmo de Adams-Bashforth para el problema de valor inicial. Para calcular las aproximaciones de la solución $y(t)$ en los puntos igualmente espaciados $t_n = t_0 + nh, n = 0,1,\ldots$, dados los puntos iniciales $y_0, y_1,\ldots y_m$ se usa la fórmula

$$y_{n+1} = y_n + h\left(b_0 \nabla^0 f_n + b_1 \nabla^1 f_n + \cdots + b_m \nabla^m f_n\right), \quad n = m, m+1,\ldots \tag{7.25}$$

Los valores iniciales $y_0, y_1,\ldots y_m$ se calculan mediante un método de un paso de orden $m+1$. En la siguiente tabla se presentan algunos valores para los coeficientes b_k.

Tabla 7.2 Valores de los coeficientes b_k.

k	0	1	2	3	4
b_k	1	$\frac{1}{2}$	$\frac{5}{12}$	$\frac{3}{8}$	$\frac{251}{720}$

www.FreeLibros.me

En la práctica, se desarrollan las diferencias en (7.25) de modo que esta ecuación se puede escribir en la forma

$$y_{n+1} = y_n + h(\beta_0 f_n + \beta_1 f_{n-1} + \cdots + \beta_m f_{n-m}), n = m, m+1, \ldots$$

Por ejemplo, cuando $m=0$, se obtiene el ya conocido método de Euler,

$$y_{n+1} = y_n + hb_0 \nabla^0 f_n, \quad n = 0, 1, 2, 3, \ldots$$

Sustituyendo los valores de las funciones, se llega a

$$y_{n+1} = y_n + hf_n, \quad n = 0, 1, 2, 3, \ldots$$

Para $m = 1$, se obtiene el método de segundo orden

$$y_{n+1} = y_n + hb_0 \nabla^0 f_n + hb_1 \nabla^1 f_n, \quad n = 1, 2, 3, 4, \ldots$$

Sustituyendo los valores de las funciones se obtiene

$$y_{n+1} = y_n + hf_n + \frac{1}{2}h(f_n - f_{n-1}), \quad n = 1, 2, 3, 4, \ldots$$

Reacomodando algebraicamente, se obtiene

$$y_{n+1} = y_n + \frac{1}{2}h(3f_n - f_{n-1}), \quad n = 1, 2, \ldots$$

Para $m = 2$, se obtiene el método de tercer orden

$$y_{n+1} = y_n + hb_0 \nabla^0 f_n + hb_1 \nabla^1 f_n + hb_2 \nabla^2 f_n, \quad n = 2, 3, 4, 5, \ldots$$

Sustituyendo los valores de las funciones se llega a

$$y_{n+1} = y_n + hf_n + \frac{1}{2}h(f_n - f_{n-1}) + \frac{5}{12}h(f_n - 2f_{n-1} + f_{n-2}), \quad n = 2, 3, 4, 5, \ldots$$

Si se hacen las operaciones algebraicas, se obtiene

$$y_{n+1} = y_n + hf_n + \frac{1}{2}hf_n - \frac{1}{2}hf_{n-1} + \frac{5}{12}hf_n - \frac{10}{12}hf_{n-1} + \frac{5}{12}hf_{n-2}, \quad n = 2, 3, 4, 5, \ldots$$

Agrupando términos, se llega a

$$y_{n+1} = y_n + \frac{23}{12}hf_n - \frac{16}{12}hf_{n-1} + \frac{5}{12}hf_{n-2}, \quad n = 2, 3, 4, 5, \ldots$$

Reacomodando algebraicamente la ecuación, al final se obtiene

$$y_{n+1} = y_n + \frac{1}{12}h(23f_n - 16f_{n-1} + 5f_{n-2}), \quad n = 2, 3, 4, 5, \ldots$$

Para $m=3$, se obtiene el método de cuarto orden

$$y_{n+1} = y_n + hb_0 \nabla^0 f_n + hb_1 \nabla^1 f_n + hb_2 \nabla^2 f_n + hb_2 \nabla^2 f_n, \quad n = 3, 4, 5, 6, \ldots$$

Sustituyendo los valores de las funciones se llega a

$$y_{n+1} = y_n + hf_n + \frac{1}{2}h(f_n - f_{n-1}) + \frac{5}{12}h(f_n - 2f_{n-1} + f_{n-2}) + \frac{3}{8}h(f_n - 3f_{n-1} + 3f_{n-2} - f_{n-3}), \quad n = 3, 4, 5, 6, \ldots$$

www.FreeLibros.me

Si se hacen las operaciones algebraicas, se obtiene

$$y_{n+1} = y_n + hf_n + \frac{1}{2}hf_n - \frac{1}{2}hf_{n-1} + \frac{5}{12}hf_n - \frac{10}{12}hf_{n-1} + \frac{5}{12}hf_{n-2} + \frac{3}{8}hf_n - \frac{9}{8}hf_{n-1} + \frac{9}{8}hf_{n-2} - \frac{3}{8}hf_{n-3}, \quad n=3,4,5,6,\ldots$$

Agrupando términos, se llega a

$$y_{n+1} = y_n + \frac{55}{24}hf_n - \frac{59}{24}hf_{n-1} + \frac{37}{24}hf_{n-2} - \frac{9}{24}hf_{n-3}, \quad n=3,4,5,6,\ldots$$

Reacomodando algebraicamente la ecuación, al final se obtiene

$$y_{n+1} = y_n + \frac{1}{24}h(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}), \quad n=3,4,\ldots$$

Con la definición de consistencia y convergencia en forma similar a como se procedió con los métodos de un paso, el siguiente teorema demuestra que, bajo ciertas condiciones, los métodos de Adams-Bashforth son convergentes y consistentes de orden $m+1$. Este resultado es sólo de importancia teórica, ya que su aplicabilidad depende de si se pueden o no determinar las constantes.

Las secciones 7.9.6, 7.9.7 y 7.9.8 contienen los métodos explícitos para los casos donde se tiene que $m=1$, $m=2$ y $m=3$, respectivamente. Para poder aplicar los métodos mencionados, se implementa una ecuación diferencial ordinaria para la que se conoce una solución analítica. De esta forma se determinan los pasos anteriores necesarios en cada caso.

Teorema 7.4 El método de Adams-Bashforth (7.25) es convergente de orden $m+1$, si $y^{(m+2)}$ está acotada en el intervalo $[t_0, b]$, y los valores iniciales $y_0, y_1, \ldots y_m$, satisfacen

$$|y(t_k) - y_k| \le Dh^{m+1}, \quad k=0,1,\ldots m$$

para algún $D \ge 0$ y $h \in (0, h_0]$ donde $h_0 > 0$. Además el error global satisface

$$|y(t_n) - y_n| \le \left(\delta + \frac{h^{m+1}|b_{m+1}|M_{m+2}}{LB_m}\right)\exp((t_n - t)LB_m) - \frac{h^{m+1}|b_{m+1}|M_{m+2}}{LB_m} \tag{7.26}$$

donde

$$\max_{t\in[t_0,b]}\left|y^{(m+2)}(t)\right| \le M_{m+2}$$

$$B_m = |\beta_0| + |\beta_1| + \cdots + |\beta_m|$$

$$\delta = \max_{0\le k\le m}|y(t_k) - y_k|$$

y L es la constante de Lipschitz para f en la segunda variable.

No existe ninguna razón para considerar la integración sobre el intervalo $[t_n, t_{n+1}]$ en la fórmula (7.23). Se puede considerar la integral en el intervalo $[t_{n-1}, t_{n+1}]$ y obtener

$$y(t_{n+1}) - y(t_{n-1}) = \int_{t_{n-1}}^{t_{n+1}} f(t, y(t))dt$$

Reemplazando $f(t, y(t))$ por un polinomio, como se hizo anteriormente, se puede deducir la fórmula

$$y_{n+1} = y_{n-1} + h\left(b_0^*\nabla^0 f_n + b_1^*\nabla^1 f_n + \cdots + b_m^*\nabla^m f_n\right) \tag{7.27}$$

con

$$b_k^* = \int_{-1}^{1}(-1)^k\binom{-s}{k}ds$$

Los métodos de este tipo reciben el nombre de *métodos de Nyström*.

www.FreeLibros.me

7.4.2 Métodos implícitos

Los métodos de Adams-Bashforth y de Nyström se llaman *métodos explícitos* o *métodos abiertos*. Las fórmulas (7.25) y (7.27) no utilizan $f(t_{n+1}, y_{n+1})$; estás fórmulas expresan explícitamente y_{n+1} en términos de $y_n, y_{n-1}, \ldots, y_{n-m}$. La naturaleza explícita de estos algoritmos proviene del hecho de usar extrapolación al integrar a $f(t, y(t))$ en el intervalo $[t_n, t_{n+1}]$. La extrapolación es, en general, menos exacta que la interpolación. Si se inpola en los puntos (t_{n-m}, f_{n-m}), $(t_{n-m+1}, f_{n-m+1}), \ldots, (t_{n-1}, f_{n-1})$, (t_n, f_n), (t_{n+1}, f_{n+1}) con $m \leq n$ se obtienen métodos implícitos.

Como se hizo en la sección anterior, se aproxima la integral (7.23) suponiendo que se construye el polinomio interpolador que pasa por los puntos (t_{n-m}, f_{n-m}), $(t_{n-m+1}, f_{n-m+1}), \ldots, (t_{n-1}, f_{n-1})$, (t_n, f_n), (t_{n+1}, f_{n+1}). Considerando a este polinomio como una aproximación a $f(t, y(t))$ en el intervalo $[t_n, t_{n+1}]$ y reemplazando (7.23) por

$$y(t_{n+1}) - y(t_n) = h\int_{-1}^{0}\left(\sum_{k=0}^{m+1}\nabla^k f_{n+1}(-1)^k\binom{-s}{k} + h^{m+2}(-1)^{m+2}\binom{-s}{m+2}f^{(m+2)}(\xi)\right)ds,$$

donde se ha efectuado el cambio de variable $t = t_{n+1} + sh$, se tiene

$$\begin{aligned} y(t_{n+1}) - y(t_n) &= h\sum_{k=0}^{m+1}\nabla^k f_{n+1}\left(\int_{-1}^{0}(-1)^k\binom{-s}{k}ds\right) + h^{m+3}\int_{-1}^{0}(-1)^{m+2}\binom{-s}{m+2}f^{(m+2)}(\xi)ds \\ &= h\sum_{k=0}^{m+1}\nabla^k f_{n+1}\left(\int_{-1}^{0}(-1)^k\binom{-s}{k}ds\right) + h^{m+3}f^{(m+2)}(\xi_n)\int_{-1}^{0}(-1)^{m+2}\binom{-s}{m+2}ds \end{aligned}$$

Si se define

$$c_k = \int_{-1}^{0}(-1)^k\binom{-s}{k}ds,$$

se obtiene

$$y(t_{n+1}) = y(t_n) + h\sum_{k=0}^{m+1}c_k\nabla^k f_{n+1} + h^{m+3}c_{m+2}f^{(m+2)}(\xi_n) \tag{7.28}$$

Definiendo el método de Adams-Moulton como

$$y_{n+1} = y_n + h\left(c_0\nabla^0 f_{n+1} + c_1\nabla^1 f_{n+1} + \cdots + c_{m+1}\nabla^{m+1}f_{n+1}\right),\ n = m+1, m+2, \ldots \tag{7.29}$$

Los valores iniciales $y_0, y_1, \ldots, y_{m+1}$ se calculan mediante un método de un paso de orden $m+2$. Los coeficientes c_k también son independientes de n. La tabla 7.3 presenta algunos valores para los coeficientes c_k.

Tabla 7.3 Valores de los coeficientes c_k.

k	0	1	2	3	4
c_k	1	$-\frac{1}{2}$	$-\frac{1}{12}$	$-\frac{1}{24}$	$-\frac{19}{720}$

Expandiendo las diferencias en (7.29) se obtiene

$$y_{n+1} = y_n + h(\gamma_{-1}f_{n+1} + \gamma_0 f_n + \gamma_1 f_{n-1} + \cdots + \gamma_m f_{n-m}), \quad n = m, m+1, \ldots \tag{7.30}$$

En general, este método es implícito, por lo que es necesario resolver esta ecuación para y_{n+1} mediante un método iterativo.

Suponiendo que $y_{m+1}^{(i)}$ es la i-ésima iteración en este proceso, entonces (7.30) se puede escribir en forma conveniente como una iteración de la forma

www.FreeLibros.me

$$w_{i+1} = F(w_i),$$

donde F es una función dada y $w_i = y_{n+1}^{(i)}$. Los valores de $y_{n-m}, y_{n-m-1}, \ldots, y_n$ permanecen sin cambio durante este proceso iterativo para calcular y_{i+1}. A fin de calcular una buena aproximación $y_{n+1}^{(0)}$ de y_{n+1}, se usa una de las fórmulas explícitas de Adams-Bashforth desarrolladas en la sección anterior. Estas ideas definen a los *métodos de predictor-corrector* tipo Adams para el problema de valor inicial.

Para calcular las aproximaciones a la solución $y(t)$ en los puntos igualmente espaciados $t_k = t_0 + kh$, $k = 0, 1, \ldots$, dadas las aproximaciones iniciales $y_0, y_1, \ldots, y_m$, se tiene para $n = m, m+1, \ldots$

$$\text{Predictor} \quad y_{n+1}^{(0)} = y_n + h[\beta_0 f_n + \beta_1 f_{n-1} + \cdots + \beta_m f_{n-m}] \tag{7.31}$$

$$\text{Corrector} \quad y_{n+1}^{(i+1)} = y_n + h\left[\gamma_{-1} f_{n+1}^{(i)} + \gamma_0 f_n + \cdots + \gamma_m f_{n-m}\right] \to i = 0, 1, \ldots, I-1 \tag{7.32}$$

$$y_{n+1} = y_{n+1}^{(I)}$$

donde

$$f_{n+1}^{(i)} = f\left(t_{n+1}, y_{n+1}^{(i)}\right), \qquad f_k = f(t_k, y_k)$$

La ecuación (7.31) se llama el *predictor*, y (7.32) se llama el *corrector*. Para cada paso se corrige I veces. Se puede simplificar la iteración del corrector notando que

$$\text{Corrector I} \quad y_{n+1}^{(i+1)} = y_n^{(i)} + h\gamma_{-1}\left[f_{n+1}^{(i)} - f_{n+1}^{(i-1)}\right], \quad i = 1, 2, \ldots, I-1 \tag{7.33}$$

por lo que se usa la fórmula (7.32) para la primera corrección y la fórmula (7.33) para las correcciones siguientes.

Para $m = 0$, el método predictor-corrector se reduce al método de segundo orden

$$\text{P: } y_{n+1}^{(0)} = y_n + hf_n$$

$$\text{C: } y_{n+1}^{(1)} = y_n + hc_0\nabla^0 f_{n+1} + hc_1\nabla^1 f_{n+1}, \quad n = 1, 2, 3, 4, \ldots$$

$$\text{C: } y_{n+1}^{(1)} = y_n + hf_{n+1} - \frac{1}{2}h(f_{n+1} - f_n), \quad n = 1, 2, 3, 4, \ldots$$

$$\text{C: } y_{n+1}^{(1)} = y_n + \frac{1}{2}h\left(f_{n+1}^{(0)} + f_n\right), \quad n = 1, 2, 3, 4, \ldots$$

$$\text{C I: } y_{n+1}^{(i+1)} = y_n^{(i)} + \frac{1}{2}h\left(f_{n+1}^{(i)} - f_{n+1}^{(i-1)}\right), \quad i = 1, 2, \ldots, I-1$$

La fórmula para el corrector recibe el nombre de *regla trapezoidal*.

Para $m = 1$ se tiene el método de tercer orden

$$\text{P: } y_{n+1}^{(0)} = y_n + \frac{h}{2}(3f_n - f_{n-1})$$

$$\text{C: } y_{n+1}^{(1)} = y_n + hc_0\nabla^0 f_{n+1} + hc_1\nabla^1 f_{n+1} + hc_2\nabla^2 f_{n+1}, \quad n = 2, 3, 4, \ldots$$

$$\text{C: } y_{n+1}^{(1)} = y_n + hf_{n+1} - \frac{1}{2}h(f_{n+1} - f_n) - \frac{1}{12}h(f_{n+1} - 2f_n + f_{n-1}), \quad n = 2, 3, 4, \ldots$$

$$\text{C: } y_{n+1}^{(1)} = y_n + \frac{1}{12}h\left(5f_{n+1}^{(0)} + 8f_n - f_{n-1}\right), \quad n = 2, 3, 4, \ldots$$

$$\text{C I: } y_{n+1}^{(i+1)} = y_n^{(i)} + \frac{5}{12}h\left(f_{n+1}^{(i)} - f_{n+1}^{(i-1)}\right), \quad i = 1, 2, \ldots, I-1$$

Con $m = 2$, se obtiene el método de cuarto orden

$$\text{P: } y_{n+1}^{(0)} = y_n + \frac{h}{12}(23f_n - 16f_{n-1} + 5f_{n-2})$$

www.FreeLibros.me

$$\text{C: } y_{n+1}^{(1)} = y_n + hc_0\nabla^0 f_{n+1} + hc_1\nabla^1 f_{n+1} + hc_2\nabla^2 f_{n+1} + hc_3\nabla^3 f_{n+1}, \quad n = 2, 3, 4, \ldots$$

$$\text{C: } y_{n+1}^{(1)} = y_n + hf_{n+1} - \frac{1}{2}h(f_{n+1} - f_n) - \frac{1}{12}h(f_{n+1} - 2f_n + f_{n-1}) - \frac{1}{24}h(f_{n+1} - 3f_n + 3f_{n-1} - f_{n-2}), \quad n = 2, 3, 4, \ldots$$

$$\text{C: } y_{n+1}^{(1)} = y_n + \frac{1}{24}h\left(9f_{n+1}^{(0)} + 19f_n - 5f_{n-1} + f_{n-2}\right), \quad n = 2, 3, 4, \ldots$$

$$\text{C I: } y_{n+1}^{(i+1)} = y_n^{(i)} + \frac{9h}{24}\left[f_{n+1}^{(i)} - f_{n+1}^{(i-1)}\right], \quad i = 1, 2, \ldots, I-1$$

Los valores iniciales $y_0, y_1, \ldots, y_m$ se deben calcular mediante un método de un paso de orden $m+2$ a fin de mantener la precisión como lo muestra el siguiente teorema.

Teorema 7.5 El método de Adams-Moulton (7.32)-(7.33) con el corrector satisfecho exactamente en cada paso es convergente de orden $m+2$ si $y^{(m+3)}$ está acotada en el intervalo $[t_0, b]$ y los valores iniciales $y_0, y_1, \ldots, y_m$ satisfacen

$$|y(t_k) - y_k| \le Dh^{m+2}, \quad k = 0, 1, \ldots, m$$

para algún $D \ge 0$ y $h \in (0, h_0]$ donde $h_0 > 0$. Además, el error global satisface

$$|y(t_n) - y_n| \le \left(\delta + \frac{h^{m+2}|c_{m+2}|M_{m+3}}{LC_m}\right)\exp\left(\frac{(t_n - t)LC_m}{1 - hL|\gamma_{-1}|}\right) - \frac{h^{m+2}|c_{m+2}|M_{m+3}}{LC_m} \tag{7.34}$$

siempre que $h < \dfrac{1}{L|\gamma_{-1}|}$, $\max\limits_{t\in[t_0,\, b]}\left|y^{(m+3)}(t)\right| \le M_{m+3}$, $C_m = |\gamma_{-1}| + |\gamma_0| + |\gamma_1| + \cdots + |\gamma_m|$, $\delta = \max\limits_{0\le k\le m}|y(t_k) - y_k|$ y L es la constante de Lipschitz para f en la segunda variable. Los métodos de Adams-Bashforth, Nyström y de Adams-Moulton aquí desarrollados se llaman, en forma genérica, *métodos de* $m+1$ *pasos*.

7.4.3 Iteración con el corrector

Verificando que el proceso anterior (7.32), para resolver el corrector, es convergente para h pequeña. Se escribe (7.30) en la forma

$$y_{n+1}^{(i+1)} = y_n + h(\gamma_0 f_n + \gamma_1 f_{n-1} + \cdots + \gamma_m f_{n-m}) + h\gamma_{-1} f\left(t_{n+1}, y_{n+1}^{(i)}\right), \tag{7.35}$$

por la cual al definir $w_i = y_{n+1}^{(i)}$ se tiene que $w_{i+1} = F(w_i)$. La sucesión $\{w_i\}_{i=0}^{\infty}$ convergerá si F satisface una condición de Lipschitz en un intervalo apropiado. En este caso se toma el intervalo como todos los números reales, buscando una constante L_F tal que $L_F < 1$ y

$$|F(w) - F(z)| \le L_F|w - z| \tag{7.36}$$

para todo w, z real. De (7.35) se obtiene

$$|F(w) - F(z)| = |h\gamma_{-1} f(t_{n+1}, w) - h\gamma_{-1} f(t_{n+1}, z)| = hL|\gamma_{-1}||w - z|$$

donde L es la constante de Lipschitz para f en la segunda variable. Se obtiene la condición para (7.36) con $L_F < 1$ si

$$h < \frac{1}{L|\gamma_{-1}|}$$

Se puede observar que esta condición se satisface siempre que se satisfagan las condiciones del Teorema 7.4.

www.FreeLibros.me

7.4.4 Estimación del error de truncamiento

Si se elige el predictor del mismo orden que el del corrector, es posible estimar el error de truncamiento en cada paso. Esto nos lleva a que la aplicación del corrector sea necesaria solamente una vez con tal de que *h* sea suficientemente pequeña. Considerando las fórmulas

$$\text{Predictor} \quad y_{n+1}^{(0)} = y_n + h[\beta_0 f_n + \beta_1 f_{n-1} + \cdots + \beta_m f_{n-m} + \beta_{m-1} f_{n-m-1}] \tag{7.37}$$

$$\text{Corrector} \quad y_{n+1}^{(i+1)} = y_n + h\left[\gamma_{-1} f_{n+1}^{(i)} + \gamma_0 f_n + \gamma_1 f_{n-1} + \cdots + \gamma_m f_{n-m}\right] \rightarrow i = 0, 1, \ldots, I-1, \tag{7.38}$$

resulta

$$y_{n+1} = y_{n+1}^{(I)}$$

A diferencia de los métodos anteriores, se necesita un valor inicial extra, pero estas fórmulas se implementan en forma idéntica. Así, si se tiene que,

$$y(t_{n+1}) = y(t_n) + h\sum_{j=0}^{m+1} b_j \nabla^j f(t_n, y(t_n)) + h^{m+3} b_{m+2}\, y^{(m+3)}(\eta_n), \tag{7.39}$$

$$y(t_{n+1}) = y(t_n) + h\sum_{j=0}^{m+1} c_j \nabla^j f(t_{n+1}, y(t_{n+1})) + h^{m+3} c_{m+2}\, y^{(m+3)}(\xi_n) \tag{7.40}$$

donde η_n y ξ_n son puntos intermedios, suponiendo que

$$y^{(n+3)}(\eta_n) \approx y^{(n+3)}(\xi_n)$$

y definiendo

$$T_n = h^{m+2} c_{m+2}\, y^{(n+3)}(\xi_n) \tag{7.41}$$

como el error de truncamiento para (7.40), restando (7.39) de (7.40), se obtiene

$$\left(\frac{b_{m+2}}{c_{m+2}} - 1\right) T_n \approx \sum_{j=0}^{m+1} c_j \nabla^j f(t_{n+1}, y(t_{n+1})) - \sum_{j=0}^{m+1} b_j \nabla^j f(t_n, y(t_n))$$

Usando las aproximaciones

$$\nabla^j f(t_n, y(t_n)) \approx \nabla^j f_n, \qquad \nabla^j f(t_{n+1}, y(t_{n+1})) \approx \nabla^j f_n^{(0)}$$

se obtiene

$$\begin{aligned} T_n &\approx \left(\frac{c_{m+2}}{b_{m+2} - c_{m+2}}\right) \sum_{j=0}^{m+1} c_j \nabla^j f_{n+1}^{(0)} - \sum_{j=0}^{m+1} b_j \nabla^j f_n \\ &= \frac{1}{h}\left(\frac{c_{m+2}}{b_{m+2} - c_{m+2}}\right)\left(y_{n+1}^{(1)} - y_{n+1}^{(0)}\right) \end{aligned} \tag{7.42}$$

Se puede usar esta estimación para el error de truncamiento en cada paso. Si se hace una corrección en cada paso, la estimación del error es

$$|\rho_n| = \left|y_{n*1}^{(2)} - y_{n*1}^{(1)}\right| \leq hL|\gamma_{-1}|\left|y_{n*1}^{(1)} - y_{n*1}^{(0)}\right|$$

Ahora, de (7.42) y (7.41) se obtiene

www.FreeLibros.me

$$hL|\gamma_{-1}||y_{n*1}^{(1)} - y_{n*1}^{(0)}| \approx h^2 L|\gamma_{-1}|\left|\frac{b_{m+2} - c_{m+2}}{c_{m+2}}\right||T_n|$$
$$\leq h^{m+4} L|\gamma_{-1}||b_{m+2} - c_{m+2}|M_{m+3}$$

donde

$$M_{m+3} \geq \max_{t\in[t_0,b]} |y^{(m+3)}(t)|$$

por lo que una cota superior para ρ_n es

$$\bar{\rho} = h^{m+4} L|\gamma_{-1}||b_{m+2} - c_{m+2}|M_{m+3}$$

y $\dfrac{\bar{\rho}}{h}$ es pequeño comparado con

$$h^{m+2}|c_{m+2}|M_{m+2} > |T_n|$$

si

$$h << \frac{1}{L|\gamma_{/1}|}\frac{|c_{m+2}|}{|b_{m+2} - c_{m+2}|} \tag{7.43}$$

Si se satisface esta condición, no habrá ventaja al utilizar el corrector más de una vez.

EJEMPLO 7.16

Considerar el siguiente método predictor-corrector de cuarto orden

$$\text{Predictor} \quad y_{n+1}^{(0)} = y_n + \frac{h}{24}[55 f_n - 59 f_{n-1} + 37 f_{n-2} - 9 f_{n-3}]$$
$$\text{Corrector} \quad y_{n+1} = y_n + \frac{h}{24}\left[9 f_{n+1}^{(0)} + 19 f_n - 5 f_{n-1} + f_{n-2}\right]$$

En este caso $m = 2$, $b_{m+2} = b_4 = \dfrac{251}{720}$ y $c_{m+2} = c_4 = -\dfrac{19}{720}$. El error de truncamiento del predictor es

$$h^{m+2} b_{m+2} y^{(m+3)}(\eta_n) = \frac{251}{720} h^4 y^{(5)}(\eta_n)$$

y el del corrector

$$T_n = h^{m+2} c_{m+2} y^{(m+3)}(\xi_n) = -\frac{19}{720} h^4 y^{(5)}(\xi_n)$$

De (7.42), se obtiene

$$T_n \approx \frac{1}{h}\left(\frac{c_4}{b_4 - c_4}\right)\left(y_{n+1}^{(1)} - y_{n+1}^{(0)}\right) = \frac{1}{h}\left(-\frac{19}{270}\right)\left(y_{n+1}^{(1)} - y_{n+1}^{(0)}\right)$$

Si se desea efectuar sólo una iteración del corrector, de (7.43) con $\gamma_{-1} = \dfrac{9}{24}$ se requiere que

$$h << \frac{8}{3L}\frac{19}{720} \approx \frac{0.2}{L}$$

www.FreeLibros.me

7.5 Métodos multipaso lineales

Dado el problema de valor inicial

$$y' = f(t, y),\ y(t_0) = y_0, \quad t \in [t_0, b]$$

un método multipaso de r pasos para la ecuación diferencial es un método de la forma

$$y_{n+r} + a_{r-1}y_{n+r-1} + \cdots + a_0 y_n = h\phi(t_n, y_{n+r}, y_{n+r-1}, \ldots, y_n; h), \quad n = r, r+1, \ldots \tag{7.44}$$

Si

$$\phi(t_n, y_{n+r}, y_{n+r-1}, \ldots, y_n; h) = b_r f(t_{n+r}, y_{n+r}) + b_{r-1} f(t_{n+r-1}, y_{n+r-1}) + \cdots + b_0 f(t_n, y_n),$$

se dice que el método es multipaso lineal. Estos métodos también se conocen como *métodos multipaso lineales de Newton-Cotes*. Esta sección se enfoca exclusivamente a los métodos multipaso lineales. Si ϕ no depende de y_{n+r} el método se dice que es explícito. Ejemplos de métodos *multipaso lineales explícitos* son los de *Adams-Bashforth* y Nyström, mientras que los de *Adams-Moulton* son métodos multipaso lineales implícitos.

Un tipo de métodos multipaso lineales se obtiene, en general, considerando la ecuación diferencial

$$y' = f(t, y),\ y(t_0) = y_0, t \in [t_0, b]$$

e integrándola en el intervalo $[t_{n-p}, t_{n+q}]$ para obtener

$$y(t_{n+q}) - y(t_{n-p}) = \int_{t_{n-p}}^{t_{n+q}} f(t, y(t))dt$$

y reemplazando $f(t, y(t))$ por un polinomio interpolador en los $m+1$ puntos (t_{r-m}, f_{r-m}), (t_{r-m+1}, f_{r-m+1}), $\ldots$, (t_{r-1}, f_{r-1}), (t_r, f_r), donde $r = n$ o $r = n+1$. La primera elección conduce a un método explícito, y la segunda lleva a un método implícito.

Esta fórmula, con $r = n$, $p = 0$, $q = 1$, $m = 0$. Se reduce al método de Euler. Con $r = n$, $p = 1$, $q = 1$ y $m = 0$, se obtiene el método del punto medio. Tomando $r = n$, $p = 0$ y $q = 1$ se obtienen los métodos Adams-Bashforth, mientras que con $r = n + 1$, $p = 0$ y $q = 1$ se obtienen los métodos de Adams-Moulton. Los métodos de Nyström resultan cuando $r = n$, $p = 1$ y $q = 1$. El método de Milne se obtiene al tomar $r = n$, $p = 3$, $q = 1$ y $m = 3$. Con $r = n + 1$, $p = 2$ y $q = 1$ se obtienen los métodos de Milne-Simpson. Se puede observar que con esta forma de generalizar se consigue una gran herramienta para generar nuevos métodos.

Una segunda forma de construir métodos multipaso lineales es aproximar la derivada

$$y' = f(t, y),\ y(t_0) = y_0, t \in [t_0, b]$$

mediante diferencias finitas. La tabla 7.4 muestra algunas aproximaciones obtenidas en el capítulo 6.

Tabla 7.4 Métodos multipaso lineales.

n	1	2	3
$t = t_0$	$\frac{y_1 - y_0}{h}$	$\frac{-y_2 + 4y_1 - 3y_0}{2h}$	$\frac{-11y_0 + 18y_1 - 9y_2 + 2y_3}{6h}$
$t = t_1$	$\frac{y_1 - y_0}{h}$	$\frac{y_2 - y_0}{2h}$	$\frac{-5y_0 + 6y_1 - 3y_2 + 2y_3}{6h}$
$t = t_2$		$\frac{y_0 - 4y_1 + 3y_2}{2h}$	$\frac{y_0 - 6y_1 + 3y_2 + 2y_3}{6h}$

(*continúa*)

www.FreeLibros.me

Tabla 7.4 Métodos multipaso lineales (*continúa*).

n	1	2	3
$t=t_3$			$\dfrac{-2y_0+9y_1-18y_2+11y_3}{6h}$
Orden	$O(h)$	$O(h^2)$	$O(h^3)$

A partir de dos métodos de orden h, por ejemplo del método de Euler y del método de Euler implícito, se tiene

$$y_{i+1}=y_i+hf(t_i,y_i)$$

y

$$y_{i+1}=y_i+hf(t_{i+1},y_{i+1})$$

Si se consideran las aproximaciones de segundo orden, se tienen tres aproximaciones de la siguiente forma

$$y_{i+2}=4y_{i+1}+3y_i-2hf(t_i,y_i)$$

$$y_{i+2}=y_i+2hf(t_{i+1},y_{i+1})$$

$$y_{i+2}=\frac{4}{3}y_{i+1}-y_i-\frac{2}{3}f(t_{i+2},t_{i+2})$$

En el caso anterior, la segunda aproximación se conoce como el *método del punto medio*. Existe otra idea para generar este tipo de métodos; por ejemplo, considerando la fórmula

$$y_{n+r}+a_{r-1}y_{n+r-1}+\cdots+a_0y_n=h(b_rf(t_{n+r},y_{n+r})+b_{r-1}f(t_{n+r-1},y_{n+r-1})+\cdots+b_0f(t_n,y_n)),\quad n=r,r+1,\ldots \tag{7.45}$$

La idea fundamental es insertar la solución exacta de la ecuación diferencial en (7.45) y expandir las series de Taylor correspondientes alrededor del mismo punto, para así poder determinar los coeficientes $a_{r-1},\ldots,a_0,b_r,b_{r-1},\ldots,b_0$.

Procediendo con un ejemplo, la fórmula considerada es

$$y_{n+2}+a_1y_{n+1}+a_0y_n-h[b_1f_{n+1}+b_0f_n]=0$$

Al sustituir la solución de la ecuación diferencial se obtiene la fórmula para el error

$$y(t_{n+2})+a_1y(t_{n+1})+a_0y(t_n)-h[b_1f(t_{n+1},y(t_{n+1}))+b_0f(t_n,y(t_n))]=hE(t_{n+2}) \tag{7.46}$$

La expansión en series de Taylor para varios términos es

$$y(t_{n+2})=y(t_n)+2hy'(t_n)+\frac{4h^2}{2}y''(t_n)+\frac{8h^3}{6}y'''(t_n)+\frac{16h^4}{24}y^{(iv)}(t_n)+\cdots$$

$$y(t_{n+1})=y(t_n)+hy'(t_n)+\frac{h^2}{2}y''(t_n)+\frac{h^3}{6}y'''(t_n)+\frac{h^4}{24}y^{(iv)}(t_n)+\cdots$$

$$y(t_n)=y(t_n)$$

$$f(t_{n+1},y(t_{n+1}))=y'(t_{n+1})=y'(t_n)+hy''(t_n)+\frac{h^2}{2}y'''(t_n)+\frac{h^3}{6}y^{(iv)}(t_n)+\cdots$$

$$f(t_n,y(t_n))=y'(t_n)$$

www.FreeLibros.me

Sustituyendo estas expresiones en (7.46), y agrupando los coeficientes, se tiene que la ecuación (7.46) se transforma en

$$\begin{aligned}&(1+a_1+a_0)y(t_n)+(2+a_1-b_1-b_0)hy'(t_n)+\left(2+\frac{1}{2}a_1-b_1\right)h^2y''(t_n)\\&+\left(\frac{4}{3}+\frac{1}{6}a_1-\frac{1}{2}b_1\right)h^3y'''(t_n)+\left(\frac{2}{3}+\frac{1}{24}a_1-\frac{1}{6}b_1\right)h^4y^{(iv)}(t_n)\\&+\left(\frac{4}{15}+\frac{1}{120}a_1-\frac{1}{24}b_1\right)h^5y^{(v)}(t_n)+\left(\frac{4}{45}+\frac{1}{720}a_1-\frac{1}{120}b_1\right)h^6y^{(vi)}(t_n)+\cdots\\&=hE(t_{n+2})\end{aligned} \quad \textbf{(7.47)}$$

Los términos con a_r y b_r en (7.47) se escogen de tal forma que los coeficientes de orden menor que h sean cero. Por lo que el coeficiente de $y(t_n)$ será cero si $1+a_1+a_0=0$, y así sucesivamente. De esta forma sólo se pueden satisfacer las primeras cuatro ecuaciones simultáneas

$$1+a_1+a_0=0 \qquad 2+a_1-b_1-b_0=0$$

$$2+\frac{a_1}{2}-b_1=0 \qquad \frac{4}{3}+\frac{a_1}{6}-\frac{b_1}{2}=0$$

Estas ecuaciones tienen soluciones

$$a_0=-5 \qquad a_1=4$$

$$b_0=2 \qquad b_1=4$$

y con estos coeficientes se obtiene un error de orden h^3 como,

$$\begin{aligned}E(t_{n+2})=&\left(\frac{2}{3}+\frac{1}{24}a_1-\frac{1}{6}b_1\right)h^3y^{(iv)}(t_n)\\&+\left(\frac{4}{15}+\frac{1}{120}a_1-\frac{1}{24}b_1\right)h^4y^{(v)}+\left(\frac{4}{45}+\frac{1}{720}a_1-\frac{1}{120}b_1\right)h^5y^{(vi)}(t_n)+\cdots\end{aligned}$$

Así, el esquema numérico se reduce a

$$y_{n+2}=-4y_{n+1}+5y_n+h[4f_{n+1}+2f_n]$$

EJEMPLO 7.17

Considerando la aproximación $\dfrac{-5y_0+6y_1-3y_2+2y_3}{6h}$ a $y'(t_1)$. Se tiene que

$$y'(t_1)=f(t_1,y(t_1))$$

Por tanto,

$$\frac{-5y_0+6y_1-3y_2+2y_3}{6h}\approx f(t_1,y(t_1))$$

Despejando y_3 se llega a

$$y_3\approx\frac{3}{2}y_2-3y_1+\frac{5}{2}y_0+3hf(t_1,y(t_1))$$

Así, el método se define como

www.FreeLibros.me

$$y_{i+3} = \frac{3}{2} y_{i+2} - 3y_{i+1} + \frac{5}{2} y_i + 3hf(t_{i+1}, y_{i+1})$$

Este método se dice que es de tres pasos, y es un método multipaso lineal explícito.

La sección 7.9.9 de este capítulo proporciona el código desarrollado en Matlab de uno de los esquemas numéricos de los métodos multipaso. El resto de ellos se pueden implementar siguiendo la misma lógica de programación.

7.6 Consistencia, convergencia y estabilidad de los métodos multipaso

Al igual que con los métodos de un paso, se consideran tres propiedades de los métodos multipaso para evaluar su desempeño. Estas propiedades son la consistencia, la convergencia y la estabilidad. Para comenzar, se considera la consistencia de un método numérico multipaso.

7.6.1 Consistencia

Inicialmente se define el error de truncamiento del método multipaso. Sea $y(t)$ la solución exacta del problema de valor inicial

$$y' = f(t, y) \text{ con } y(t_0) = y_0$$

El error de truncamiento del método multipaso lineal dado por

$$y_{n+r} + a_{r-1} y_{n+r-1} + \cdots + a_0 y_n = h(b_r f(t_{n+r}, y_{n+r}) + b_{r-1} f(t_{n+r-1}, y_{n+r-1}) + \cdots + b_0 f(t_n, y_n)),$$
$$n = r, r+1, \ldots \tag{7.48}$$

está definido como

$$\tau(t; h) = \frac{1}{h}(y(t+rh) + a_{r-1} y(t+(r-1)h) + \cdots + a_0 y(t))$$
$$-(b_r f(t+rh, y(t+rh)) + b_{r-1} f(t+(r-1)h, y(t+(r-1)h)) + \cdots + b_0 f(t, y(t)))$$

Suponiendo que los valores iniciales $y_0, y_1, \ldots, y_{r-1}$ se aproximan a los valores exactos $y(t_0), y(t_1), \ldots,$ $y(t_{r-1})$, se dice que el método multipaso (7.48) es consistente si

a) $\lim\limits_{h \to 0} \tau(t; h) = 0$

b) $\lim\limits_{h \to 0} |y_i - y(t_i)| = 0,\ i = 0, 1, \ldots, r-1$

Definición 7.3 El esquema (7.48) es consistente de orden p si existe $M \geq 0, h_0 > 0$ y un entero positivo p tal que

$$\sup_{t_0 \leq t \leq b} |\tau(t; h)| \leq Mh^p, \text{ para toda } h \text{ en } (0, h_0]$$

Estas definiciones sólo son una generalización de las definiciones del error de truncamiento y consistencia para métodos de un paso dada en la sección 7.3. También se puede observar, por (7.24), que el método de Adams-Bashforth es consistente, de orden $m + 1$, y de (7.28), que el método de Adams-Moulton es consistente de orden $m + 2$. En general, si se definen para (7.48) las constantes

www.FreeLibros.me

$$C_0 = 1 + a_{r-1} + a_{r-2} + \cdots + a_0,$$
$$C_1 = r + (r-1)a_{r-1} + (r-2)a_{r-2} + \cdots + a_1 - \sum_{j=0}^{r} b_j$$
$$\vdots$$
$$C_p = \frac{1}{p!}\left[r^p + (r-1)^p a_{r-1} + (r-2)a_{r-2} + \cdots + a_1\right] - \frac{1}{(p-1)!}\sum_{j=0}^{r} j^{p-1} b_j$$

se puede demostrar que el esquema es consistente si y sólo si

$$C_0 = C_1 = 0$$

y es consistente de orden p si

$$C_0 = C_1 = \cdots = C_p = 0$$

siempre que se satisfaga la condición

$$\lim_{h\to 0}|y_i - y(t_i)| = 0,\ i = 0, 1, \ldots, r-1$$

EJEMPLO 7.18

Considerar la ecuación $y_{n+2} = 4y_{n+1} - 3y_n - 2hf_n$. En este caso se tiene que

$$C_0 = 1 - 4 + 3 = 0$$
$$C_1 = 2 - 4 - 2 = 0$$
$$C_2 = \frac{1}{2!}(2^2 - 4) - 0 = 0$$
$$C_3 = \frac{1}{3!}[2^3 - 4] - 0 \neq 0$$

por lo que el esquema es consistente de segundo orden.

EJEMPLO 7.19

Para el método desarrollado en la sección 7.5, se tiene la ecuación

$y_{n+2} = -4y_{n+1} + 5y_n + h[4f_{n+1} + 2f_n]$, con solución $a_0 = -5,\ a_1 = 4,\ b_0 = 2,\ b_1 = 4$ y

$$C_0 = 1 + 4 - 5 = 0$$
$$C_1 = 2 + 4 - (2 + 4) = 0$$
$$C_2 = \frac{1}{2!}(2^2 + 4) - \frac{1}{1!}(0\cdot 2 + 1\cdot 4) = 0$$
$$C_3 = \frac{1}{3!}[2^3 + 4] - \frac{1}{2!}(0^2\cdot 2 + 1^2\cdot 4) = 0$$
$$C_4 = \frac{1}{4!}[2^4 + 4] - \frac{1}{3!}(0^2\cdot 2 + 1^2\cdot 4) \neq 0$$

Por lo que se tiene que el esquema es de tercer orden, como se estableció en la sección 7.5.

www.FreeLibros.me

7.6.2 Convergencia

Para establecer las condiciones bajo las cuales este tipo de métodos converge, se comienza con la definición de convergencia en forma similar a la de los métodos de un paso.

Definición 7.4 El esquema numérico (7.48) es convergente si al aplicarlo al problema de valor inicial $y' = f(t, y)$, $y(t_0) = y_0$, $t \in [t_0, b]$, con los valores iniciales que satisfacen

$$\lim_{h \to 0} y_n = y_0,\ n = 0, 1, \ldots, r-1,$$

la solución de (7.48) satisface

$$\lim_{\substack{n \to 0 \\ t_n = t \text{ fijo}}} y_n = y(t)$$

para toda $t \in [t_0, b]$.

Los teoremas 7.4 y 7.5 afirman la convergencia de los métodos de Adams-Bashforth y de Adams-Moulton, respectivamente. La demostración de convergencia para un método numérico multipaso es, en general, más complicada que las demostraciones de consistencia o estabilidad. Afortunadamente se tiene el siguiente teorema:

Teorema 7.6 Un esquema numérico consistente y estable, como (7.48), aplicado al problema de valor inicial $y' = f(t, y)$, $y(t_0) = y_0$, $t \in [t_0, b]$, donde f es continua y satisface una condición de Lipschitz en la segunda variable, es convergente. De manera más general, se tiene que un esquema numérico es consistente y estable si y sólo si es convergente.

7.6.3 Estabilidad

Para esquemas de un paso, la consistencia es una condición suficiente para la convergencia. Para los métodos multipaso, esto no es necesariamente cierto, aun cuando la función $\phi(t_n, y_{n+r}, y_{n+r-1}, \ldots, y_n; h)$ en (7.44) satisfaga una constante de Lipschitz. Considerando el esquema

$$y_{n+2} = -4y_{n+1} + 5y_n + h[4f_{n+1} + 2f_n]$$

aplicado a la ecuación diferencial

$$y' = 0, t \in [0, T], y(0) = y_0,$$

el esquema se reduce a

$$y_{n+2} + 4y_{n+1} - 5y_n = 0, \tag{7.49}$$

la cual se puede resolver fácilmente, ya que es una ecuación lineal. La solución de la ecuación (7.49) se puede encontrar al sustituir una solución propuesta $y_n = Az^n$, lo cual reduce (7.49) a una ecuación polinomial auxiliar. Las raíces del polinomio auxiliar, si son distintas, dan los valores de z para los cuales y_n satisface la ecuación en diferencias. En este caso la ecuación auxiliar resultante de la sustitución es

$$Az^n(z^2 + 4z - 5) = 0 \tag{7.50}$$

La cual tiene dos soluciones principales, $z_1 = 1$ y $z_2 = -5$. La solución completa de la ecuación en diferencias es, por tanto

$$y_n = A_1 z_1{}^n + A_2 z_2{}^n = A_1(+1)^n + A_2(-5)^n$$

www.FreeLibros.me

Los valores de A_1 y A_2 se determinan a partir de las condiciones iniciales. En este caso, dentro de los límites de precisión de la computadora, A_1 puede ser aproximadamente la unidad, y A_2 aproximadamente cero. Por tanto, el primer término corresponde a la solución, pero el segundo es un término falso que tiene propiedades no satisfactorias. Estos términos falsos siempre surgen cuando se emplean los métodos multipaso como un intento por incrementar la precisión del esquema. En este caso la magnitud de los términos falsos se incrementa por un factor de 5 en cada etapa, por lo que si A_2 es un término pequeño, al final dominará la solución verdadera. Por ejemplo, después de sólo 10 pasos, el término $(z_2)^n$ es aproximadamente 10^7, aunque A_2 al principio sea menor que 10^{-7}. El término falso es siempre tan grande como la solución real, y puede ser el término más significativo de la solución desde este punto en adelante.

Es claro que donde se introducen en la fórmula los términos falsos, es esencial que éstos disminuyan rápidamente. La marcada inestabilidad mostrada en la ecuación anterior se puede eliminar fácilmente incluyendo el término f_{n+2}, pero sin incrementar la precisión. Esto lleva a cuatro ecuaciones con cinco incógnitas, por lo que se tienen parámetros libres que es posible seleccionar para proporcionar mejores propiedades de estabilidad. El conjunto de ecuaciones sería ahora

$$1+a_1+a_0=0$$

$$2+a_1-b_2-b_1-b_0=0$$

$$2+\frac{a_1}{2}-2b_2-b_1=0$$

$$\frac{4}{3}+\frac{a_1}{6}-2b_2-\frac{b_1}{2}=0$$

Todos los coeficientes se pueden expresar en términos de un único coeficiente, por ejemplo a_0, de manera que el valor de este coeficiente se selecciona para proporcionar estabilidad:

$$a_0=a_0 \qquad a_1=-1-a_0$$

$$b_2=\frac{5}{12}+\frac{a_0}{12} \qquad b_1=\frac{2}{3}-\frac{2}{3}a_0 \qquad b_0=-\frac{1}{12}-\frac{5}{12}a_0$$

Mediante el uso de una solución propuesta Az^n, como se realizó anteriormente, se obtiene la ecuación auxiliar

$$z^2-(1+a_0)z+a_0=0$$

la cual tiene soluciones $z=1$ y $z=a_0$. Con el objetivo de asegurar que la solución falsa no se incremente cuando n se incrementa, es necesario que $|a_0|\le 1$, por lo que hay diversas fórmulas que son estables. A continuación se dan algunos ejemplos de éstas:

a_0	-1	0	1
a_1	0	-1	-2
b_0	$\frac{1}{3}$	$-\frac{1}{12}$	$-\frac{1}{2}$
b_1	$\frac{4}{3}$	$\frac{8}{12}$	0
b_2	$\frac{1}{3}$	$\frac{5}{12}$	$\frac{1}{2}$

La primera columna es la regla de Simpson 1/3 y la segunda columna es una de las fórmulas implícitas de Adams-Moulton, que es característica en el método de Nordsieck en una forma modificada. Después de estos ejemplos, se procede a la definición formal de estabilidad de un método multipaso, la cual será una generalización de la definición de estabilidad para los métodos de un paso y a la definición, además del polinomio característico asociado al método multipaso.

www.FreeLibros.me

Definición 7.5 Un esquema numérico multipaso se dice que es estable si existe constante positiva K, independiente de n tal que si para $\delta > 0$, los dos conjuntos $\{y_0, y_1, \ldots, y_{r-1}\}$ y $\{\tilde{y}_0, \tilde{y}_1, \ldots, \tilde{y}_{r-1}\}$ satisfacen $|y_i - \tilde{y}_i| < \delta$, $i = 0, 1, \ldots, r-1$. Entonces $|y_n - \tilde{y}_n| < \delta K$, $n = r, r+1, \ldots$

Definición 7.6 El polinomio característico de (7.48) es el polinomio

$$\varphi(z) = z^r + a_{r-1}z^{r-1} + \cdots + a_1 z + a_0$$

Como $z = 1$ es una raíz de $\varphi(z)$, si el esquema es consistente y esta raíz es la principal, este polinomio es de grado r, por lo que hay $r-1$ raíces más. Éstas se llaman *raíces parásitas* y corresponden a soluciones del esquema multipaso que no se aproximan a la solución de la ecuación diferencial. En el ejemplo anterior $z_2 = -5$ es una raíz parásita y lleva a soluciones no precisas. Ahora, dado que la solución depende del comportamiento de las raíces del polinomio característico, se tiene la siguiente definición:

Definición 7.7 El esquema multipaso (7.48) satisface la condición de raíz si todos los ceros z del polinomio característico $\varphi(z)$ satisfacen

i) $|z| \le 1$
ii) $|z| = 1$, en cuyo caso z es una raíz simple de $\varphi(z)$

Se tiene que esta condición es equivalente a la estabilidad. Esto es, si un esquema numérico satisface la condición de raíz, entonces el esquema es estable. Usando la condición de cada raíz se puede comprobar cuando un esquema multipaso es estable. Los métodos Adams-Bashforth y Adams-Moulton satisfacen esta condición.

7.7 Solución numérica de sistemas de ecuaciones diferenciales ordinarias

Dado el problema de valor inicial

$$F\left(t, y, y', y'', \ldots, y^{(n)}\right) = 0 \qquad \textbf{(7.51)}$$

con las condiciones iniciales

$$\begin{aligned} y(a) &= y_0 \\ y'(a) &= y'_0 \\ &\vdots \\ y^{(n-1)}(a) &= y_0^{(n-1)} \end{aligned}$$

es posible extender los métodos de solución para el problema de valor inicial de primer orden para resolver esta ecuación.

La idea fundamental es transformar la ecuación (7.51) en un sistema de ecuaciones de primer orden mediante las transformaciones

$$\begin{aligned} x_1(t) &= y(t) \\ x_2(t) &= y'(t) \\ &\vdots \\ x_n(t) &= y^{(n-1)}(t) \end{aligned}$$

www.FreeLibros.me

para obtener

$$\begin{aligned}
x_1' &= x_2 \\
x_2' &= x_3 \\
&\vdots \\
x_{n-1}' &= x_{n-2} \\
x_n' &= y^{(n)}(t)
\end{aligned}$$

Si (7.51) se puede resolver para $y^{(n)}(t)$, esto es, $y^{(n)} = f\left(t, y, y', y'',\ldots, y^{(n-1)}\right)$, entonces (7.51) se reduce al sistema

$$\begin{aligned}
x_1' &= x_2 \\
x_2' &= x_3 \\
&\vdots \\
x_{n-1}' &= x_{n-2} \\
x_n' &= f(t, x_1, x_2, \ldots, x_n)
\end{aligned}$$

Las condiciones iniciales se transforman ahora en

$$\begin{aligned}
x_1(a) &= y_0 \\
x_2(a) &= y_0' \\
&\vdots \\
x_n(a) &= y_0{}^{(n-1)}
\end{aligned}$$

En forma general, un problema de valor inicial de orden n se puede transformar en un sistema de n ecuaciones diferenciales de primer orden con la forma

$$\begin{aligned}
x_1' &= f_1(t, x_1, x_2,\ldots, x_n) \\
x_2' &= f_2(t, x_1, x_2,\ldots, x_n) \\
&\vdots \\
x_{n-1}' &= f_{n-1}(t, x_1, x_2,\ldots, x_n) \\
x_n' &= f_n(t, x_1, x_2,\ldots, x_n)
\end{aligned} \tag{7.52}$$

sujeto a

$$\begin{aligned}
x_1(a) &= x_{10} \\
x_2(a) &= x_{20} \\
&\vdots \\
x_n(a) &= x_{n0}
\end{aligned} \tag{7.53}$$

Los métodos de solución de una ecuación diferencial ordinaria se pueden extender para incluir el caso de un sistema de ecuaciones diferenciales; de entre ellos, los más utilizados son:

1. El método de Euler
2. El método de Euler trapezoidal
3. Los métodos de Runge-Kutta

A continuación se describe el esquema numérico de cada uno de estos métodos.

www.FreeLibros.me

7.7.1 Método de Euler

Esquema numérico de Euler aplicado a un sistema de ecuaciones diferenciales ordinarias:

$$\begin{aligned} x_1' &= f_1(t, x_1, x_2, \ldots, x_n) \\ x_2' &= f_2(t, x_1, x_2, \ldots, x_n) \\ &\vdots \\ x_n' &= f_n(t, x_1, x_2, \ldots, x_n) \end{aligned}$$

donde $x_1(a) = x_{10}$, $x_2(a) = x_{20}, \ldots, x_n(a) = x_{n0}$ son las condiciones iniciales.

Así, el método se define como [Maron *et al.*, 1995]

$$\begin{aligned} x_{1,k+1} &= x_{1,k} + hf_1(t_k, x_{1,k}, x_{2,k}, \ldots, x_{n,k}) \\ x_{2,k+1} &= x_{2k} + hf_2(t_k, x_{1,k}, x_{2,k}, \ldots, x_{n,k}) \\ &\vdots \\ x_{n,k+1} &= x_{n;k} + hf_n(t_k, x_{1,k}, x_{2,k}, \ldots, x_{n,k}) \end{aligned}$$

Este esquema es general, debido a que cualquier ecuación diferencial de alto orden se puede descomponer o expresar como un sistema de ecuaciones de primer orden. El código Matlab de este método se presenta en la sección 7.9.10.

EJEMPLO 7.20

Resolver por el método de Euler la ecuación diferencial ordinaria $v''(t) + 3v'(t) + 8v(t) = 0$, con las condiciones iniciales $v(0) = v_0 = 1$ y $v'(0) = v_0' = 0$, en el intervalo $0 \leq t \leq 4$, con un paso de $h = 0.05$.

Si se hace la transformación,

$$v' = z \qquad v_0 = 1$$

$$z' = -3z - 8v \qquad z_0 = 0$$

El esquema numérico para esta ecuación queda de la siguiente manera

$$v_n = v_{n-1} + h[z_{n-1}]$$

y

$$z_n = z_{n-1} + h[-3z_{n-1} - 8v_{n-1}]$$

Los primeros dos pasos de este esquema numérico son los siguientes

$$t_1 = 0.1 \qquad v_1 = v_0 + h[z_0] = 1 + 0.05 \times (0) = 1$$

$$z_1 = z_0 + h[-3z_0 - 8v_0] = 0 + 0.5 \times (-3(0) - 8(1)) = -0.4$$

$$t_2 = 0.2 \qquad v_2 = v_1 + h[z_1] = 1 + 0.05 \times (-0.4) = 0.98$$

$$z_2 = z_1 + h[-3z_1 - 8v_1] = -0.4 + 0.05 \times (-3(-0.4) - 8(1)) = -0.74$$

Los siguientes resultados se muestran en la figura 7.3, donde se compara el resultado del método numérico con el resultado analítico de esta ecuación, el cual es $v(t) = e^{\alpha t}(\cos \beta t - (\alpha/\beta)\operatorname{sen} \beta t)$, con $\alpha = -1.5$ y $\beta = 2.3979$.

www.FreeLibros.me

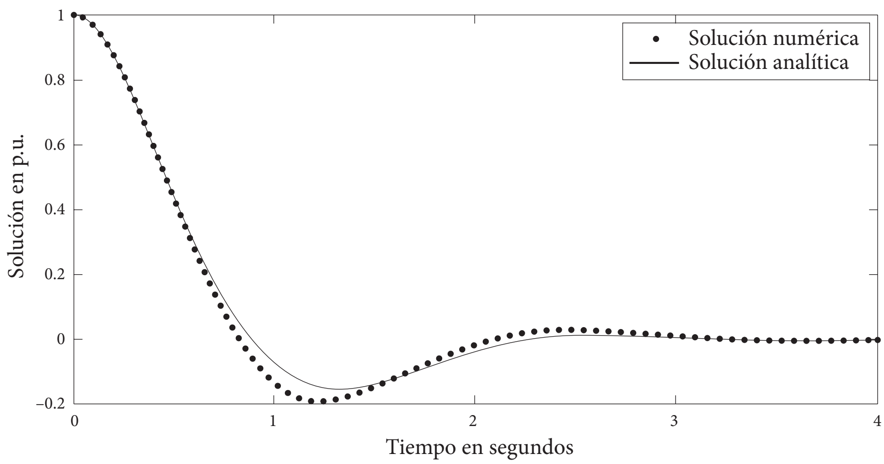


Figura 7.3 Solución numérica con $h = 0.05$ *vs.* solución analítica.

La solución numérica se puede ajustar mejor a la solución analítica si se elige un paso de integración más pequeño. Por ejemplo, con $h = 0.01$, la figura 7.4 muestra que el error en el cálculo se reduce bastante con referencia a la figura 7.3.

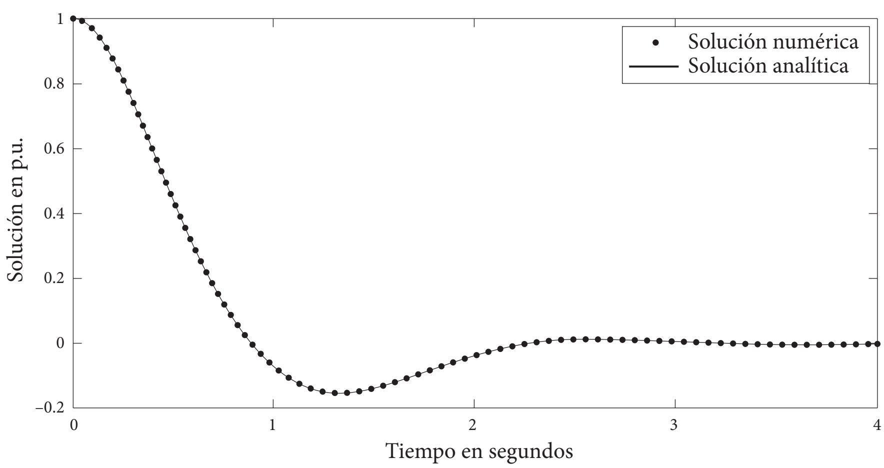


Figura 7.4 Solución numérica con $h = 0.01$ *vs.* solución analítica.

7.7.2 Método de Euler trapezoidal

Las modificaciones al esquema numérico de Euler llevan a un esquema más preciso y más estable. El esquema numérico del método de Euler trapezoidal es

$$x_{1,k+1} = x_{1,k} + \frac{h}{2}\left(f_1\left(t_{k+1}, x_{1,k+1}, x_{2,k+1}, \ldots, x_{n,k+1}\right) + f_1\left(t_k, x_{1,k}, x_{2,k}, \ldots, x_{n,k}\right)\right)$$

$$x_{2,k+1} = x_{2k} + \frac{h}{2}\left(f_2\left(t_{k+1}, x_{1,k+1}, x_{2,k+1}, \ldots, x_{n,k+1}\right) + f_2\left(t_k, x_{1,k}, x_{2,k}, \ldots, x_{n,k}\right)\right)$$

$$\vdots$$

www.FreeLibros.me

$$x_{n,k+1} = x_{n;k} + \frac{h}{2}\left(f_n(t_{k+1}, x_{1,k+1}, x_{2,k+1}, \ldots, x_{n,k+1}) + f_n(t_k, x_{1,k}, x_{2,k}, \ldots, x_{n,k})\right)$$

Analizando el esquema anterior, se puede observar que las funciones dependen de sí mismas. Por esta razón se necesita iterar hasta su convergencia. La forma de iterar conduce a la utilización del método de Gauss-Seidel que, operativamente, se puede resumir de manera fácil como sigue:

- Se conocen todas las condiciones iniciales, esto es $x_{1,k}$, $x_{2,k}$, …, $x_{n,k}$.
- Para la primera ecuación, es decir para $x_{1,k+1}$, se suponen valores para todas las variables, o lo que es lo mismo, se dan valores a $x_{1,k+1}$, $x_{2,k+1}$, …; $x_{n,k+1}$.
- Para la segunda ecuación, se utiliza el valor calculado de $x_{1,k+1}$ y los valores supuestos de $x_{2,k+1}$,…, $x_{n,k+1}$ para determinar el valor de $x_{2,k+1}$.
- Se procede con las demás ecuaciones de la misma forma, utilizando siempre el valor más actualizado de las variables.
- El proceso se detiene cuando todas las variables difieren entre sí en dos iteraciones consecutivas en un valor muy pequeño o, en su defecto, se establece un número máximo de iteraciones para detener el proceso.

Este método de Euler con la modificación de la regla trapezoidal proporciona un error de h^2, mientras que el error global proporcional del método de Euler es h. Al aplicar este método ya modificado, se debe resolver el grupo de ecuaciones en forma simultánea o implícita. Sin embargo, la ventaja de la solución implícita consiste en que el método es más estable; por esta razón permite un mayor intervalo de tiempo de integración.

EJEMPLO 7.21

Resolver por el método de Euler trapezoidal la ecuación diferencial ordinaria del ejemplo 7.20 y compararla con el resultado anterior.

La ecuación por resolver es $v''(t)+3v'(t)+8v(t)=0$, con las condiciones iniciales $v(0)=v_0=1$ y $v'(0)=v_0'=0$, en el intervalo $0 \le t \le 4$, con un paso de $h=0.05$.

Si se hace la misma transformación que en el ejemplo 7.20 se obtiene

$$v' = z \qquad v_0 = 1$$

$$z' = -3z - 8v \qquad z_0 = 0$$

El esquema numérico para esta ecuación queda de la siguiente manera

$$v_n = v_{n-1} + \frac{1}{2}h\left[(z_{n-1}) + (z_n)\right]$$

$$z_n = z_{n-1} + \frac{1}{2}h\left[(-3z_{n-1} - 8v_{n-1}) + (-3z_n - 8v_n)\right]$$

De la figura 7.5 se puede observar cómo, efectivamente, el método Euler trapezoidal converge mejor. El método desarrollado en Matlab de la técnica Euler trapezoidal se presenta en la sección 7.9.11 de este capítulo.

www.FreeLibros.me

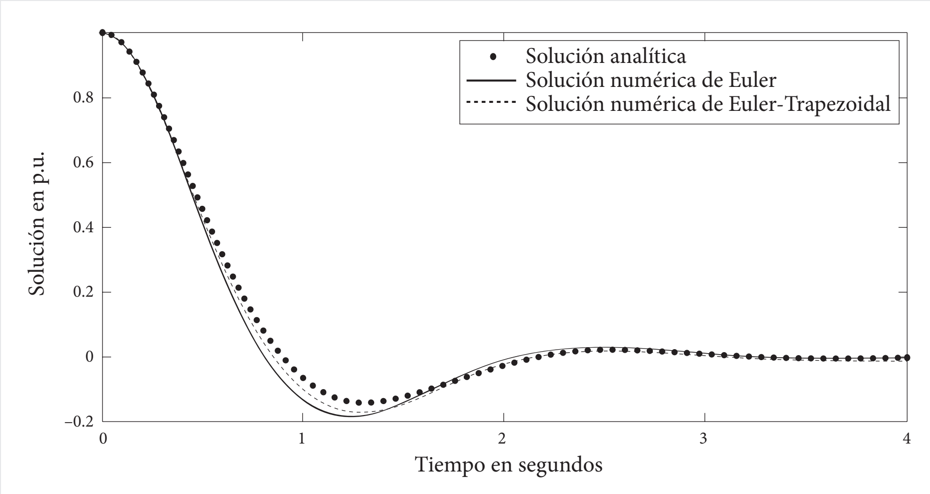


Figura 7.5 Solución numérica con $h = 0.05$ *vs*. solución analítica.

7.7.3 Métodos de Runge-Kutta

La extensión de los métodos de Runge-Kutta a las ecuaciones simultáneas es bastante fácil, y en seguida se muestran las ecuaciones correspondientes. Si se considera el problema

$$\begin{aligned} x' &= f(x, y, t) \\ y' &= g(x, y, t) \\ x_0 &= s_1 \\ y_0 &= s_2 \end{aligned}$$

entonces se puede usar el siguiente esquema computacional de segundo orden para estas ecuaciones:

$$\begin{aligned} k_1 &= hf(x_n, y_n, t_n) \\ m_1 &= hg(x_n, y_n, t_n) \\ k_2 &= hf(x_n + k_1, y_n + m_1, t_{n+1}) \\ m_2 &= hg(x_n + k_1, y_n + m_1, t_{n+1}) \end{aligned}$$

y, así, el sistema queda de la siguiente manera

$$x_{n+1} = x_n + \frac{1}{2}(k_1 + k_2)$$

$$y_{n+1} = y_n + \frac{1}{2}(m_1 + m_2)$$

El único punto que observar es que k_1 y m_1 deben calcularse antes que k_2 y m_2.

La sección 7.9.13 de este capítulo proporciona el código desarrollado en Matlab para el método de Runge-Kutta de segundo orden aplicado a sistemas de ecuaciones diferenciales ordinarias.

La extensión del esquema anterior a varias variables es de lo más sencillo y fácil de programar en una computadora. La aplicación de los métodos de Runge-Kutta de orden superior a un conjunto de ecuacio-

www.FreeLibros.me

nes diferenciales ordinarias es análogo a la aplicación del método de segundo orden. Para el caso del método de cuarto orden aplicado a un conjunto de dos ecuaciones, se tiene

$$y' = f(y, z, t)$$
$$z' = g(y, z, t)$$

con las condiciones iniciales

$$y_0 = s_1$$
$$z_0 = s_2$$

Entonces el método Runge-Kutta de cuarto orden para este conjunto es

$$k_1 = hf(y_n, z_n, t_n)$$

$$m_1 = hg(y_n, z_n, t_n)$$

$$k_2 = hf\left(y_n + \frac{k_1}{2}, z_n + \frac{m_1}{2}, t_n + \frac{h}{2}\right)$$

$$m_2 = hg\left(y_n + \frac{k_1}{2}, z_n + \frac{m_1}{2}, t_n + \frac{h}{2}\right)$$

$$k_3 = hf\left(y_n + \frac{k_2}{2}, z_n + \frac{m_2}{2}, t_n + \frac{h}{2}\right)$$

$$m_3 = hg\left(y_n + \frac{k_2}{2}, z_n + \frac{m_2}{2}, t_n + \frac{h}{2}\right)$$

$$k_4 = hf(y_n + k_3, z_n + m_3, t_n + h)$$

$$m_4 = hg(y_n + k_3, z_n + m_3, x_n + h)$$

y así sucesivamente,

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$z_{n+1} = z_n + \frac{1}{6}(m_1 + 2m_2 + 2m_3 + m_4)$$

Resulta fundamental que, si el número de ecuaciones es mayor de dos, el método de Runge-Kutta es el mismo. La sección 7.9.14 proporciona el código Matlab de este método.

EJEMPLO 7.21

Resolver por el método de Runge-Kutta de segundo orden el sistema de ecuaciones diferenciales ordinarias dado por $\mathbf{X}' + \mathbf{AX} = \mathbf{B}$. Las condiciones iniciales son $\mathbf{X}^T(0) = [0 \quad 0]$; el intervalo es $0 \le t \le 4$, con un paso de $h = \Delta t = 0.01$.

Si se tiene que

$$\mathbf{A} = \begin{bmatrix} 3 & 2 \\ 2 & 8 \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} e^{-2t} \\ e^{-3t} \end{bmatrix} \text{ y } \mathbf{X} = \begin{bmatrix} i \\ v \end{bmatrix},$$

la función de la derivada se expresa como

$$f(i, v, t) = -3i_n - 2v_n + e^{-2t}$$

www.FreeLibros.me

$$g(i,v,t)=-2i_n-8v_n+e^{-3t}$$

De aquí se obtiene que

$$k_1=hf(x_n,y_n,t_n)=h\left(-3i_n-2v_n+e^{-2(t-\Delta t)}\right)$$

$$m_1=hg(x_n,y_n,t_n)=h\left(-2i_n-8v_n+e^{-3(t-\Delta t)}\right)$$

$$k_2=hf(x_n+k_1,y_n+m_1,t_{n+1})=h\left(-3(i_n+k_1)-2(v_n+m_1)+e^{-2t}\right)$$

$$m_2=hg(x_n+k_1,y_n+m_1,t_{n+1})=h\left(-2(i_n+k_1)-8(v_n+m_1)+e^{-3t}\right)$$

Así, el esquema numérico final es

$$i_{n+1}=i_n+\frac{1}{2}(k_1+k_2)$$

$$v_{n+1}=v_n+\frac{1}{2}(m_1+m_2)$$

Si se tiene la condición inicial dada por

$$t_0=0,\ i_0=0\ \text{y}\ v_0=0,$$

el siguiente valor es

$$t_1=0.01$$

$$k_1=h\left(-3i_0-2v_0+e^{-2(0)}\right)=0.01(-3(0)-2(0)+1)=0.01$$

$$m_1=h\left(-2i_0-8v_0+e^{-3(0)}\right)=0.01(-2(0)-8(0)+1)=0.01$$

$$k_2=h\left(-3(i_0+k_1)-2(v_0+m_1)+e^{-0.02}\right)=0.01(-3(0+0.01)-2(0+0.01)+0.9802)=0.0093$$

$$m_2=h\left(-2(i_0+k_1)-8(v_0+m_1)+e^{-0.03}\right)=0.01(-2(0+0.01)-8(0+0.01)+0.9704)=0.0087$$

$$i_1=i_0+\frac{1}{2}(k_1+k_2)=0+\frac{1}{2}(0.01+0.0093)=0.0097$$

$$v_1=v_0+\frac{1}{2}(m_1+m_2)=0+\frac{1}{2}(0.01+0.0087)=0.0094$$

Los primeros resultados se muestran en la siguiente tabla.

Tabla 7.5 Resultados de aplicar el método de Runge-Kutta a un sistema de ecuaciones diferenciales ordinarias.

	k_1	m_1	k_2	m_2	i_n	v_n
$t=0.01$	0.01	0.01	0.0093	0.0087	0.0097	0.0094
$t=0.02$	0.0093	0.0088	0.0087	0.0076	0.0187	0.0175
$t=0.03$	0.0087	0.0076	0.0081	0.0066	0.0270	0.0246
$t=0.04$	0.0081	0.0066	0.0076	0.0057	0.0349	0.0308
$t=0.05$	0.0076	0.0057	0.0070	0.0048	0.0422	0.0361
$t=0.06$	0.0071	0.0049	0.0066	0.0041	0.0490	0.0405
$t=0.07$	0.0066	0.0041	0.0061	0.0034	0.0554	0.0443
$t=0.08$	0.0061	0.0035	0.0057	0.0028	0.0613	0.0475

(*continúa*)

www.FreeLibros.me

(*continuación*)

	k_1	m_1	k_2	m_2	i_n	v_n
$t = 0.09$	0.0057	0.0028	0.0053	0.0023	0.0668	0.0500
$t = 0.10$	0.0053	0.0023	0.0050	0.0018	0.0720	0.0520
$t = 0.11$	0.0050	0.0018	0.0046	0.0013	0.0768	0.0536
$t = 0.12$	0.0046	0.0014	0.0043	0.0009	0.0813	0.0548
$t = 0.13$	0.0043	0.0010	0.0040	0.0006	0.0855	0.0556

Los resultados en todo el intervalo de análisis se muestran en forma gráfica en la figura 7.6. En esta figura se puede notar que las dos variables dependientes tienden a cero a medida que el tiempo avanza. Esto se debe a que la solución del sistema de ecuaciones tiene valores característicos con parte real negativa y las entradas son exponenciales decrecientes. Por esta razón, finalmente se tenderá a cero, cosa que se muestra en los resultados numéricos.

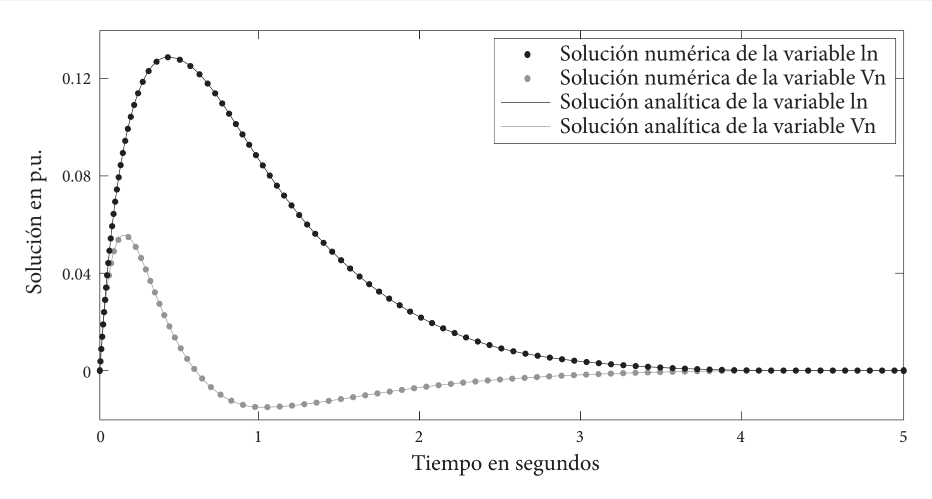


Figura 7.6 Solución numérica del sistema de ecuaciones del ejemplo 7.21, con $h = 0.01$.

7.8 Comparación de métodos

El problema de determinar los valores iniciales para esquemas multipaso se resuelve de varias formas en diferentes esquemas. El método de la expansión en series de Taylor se puede usar para algunos cuantos pasos; pero esto es más apropiado para un cálculo manual que para su uso en una computadora. Otro método alternativo puede ser el esquema de un paso Euler trapezoidal con un tamaño de paso más pequeño que el del esquema de integración principal, de manera que se pueda obtener una precisión comparable. En particular, el esquema de Runge-Kutta-Merson puede verificar que el error sea pequeño a lo largo de los pasos iniciales de la integración. El método de Nordsieck usa una serie de métodos multipaso de diferente orden, por lo que el proceso se inicia a partir de un esquema de un simple paso y se incrementa en un orden específico, utilizando una serie de diferentes fórmulas para que estén disponibles puntos adicionales. Es necesario tener cuidado al diseñar el procedimiento inicial para asegurar que la precisión sea al menos tan alta como el esquema de integración principal. Como el esquema de integración princi-

www.FreeLibros.me

pal utiliza potencias mucho más grandes de *h* en el término del error, esto nos indica que los métodos iniciales se basan en un valor más reducido que el tamaño del paso *h*.

Los métodos de Runge-Kutta se utilizan con frecuencia para resolver problemas en computadora porque no necesitan métodos especiales de inicio y, por tanto, son fáciles de programar. El método de Runge-Kutta-Merson también da un procedimiento automático para ajustar el tamaño del paso, lo que lo hace, una vez más, fácil de usar para los no especialistas. Sin embargo, hay tres desventajas en los métodos de Runge-Kutta que no se pueden vencer. Considerando que las dos características mencionadas se pueden introducir en los métodos multipaso con algún cuidado en la programación, como se hizo en el método de Nordsieck, estas desventajas hacen que los métodos multipaso sean preferibles, en general, excepto cuando la simplicidad sea la característica más importante.

Una desventaja importante de los métodos de Runge-Kutta es que la forma del término del error es extremadamente complicada, y esto no se puede superar por completo mediante el método de Merson que proporciona sólo una aproximación. Un análisis de los métodos para encontrar las cotas sobre los errores es muy difícil, considerando que existe una fórmula sencilla para los métodos multipaso, con tal de que la derivada apropiada se pueda calcular. Otra característica inconveniente de los métodos de Runge-Kutta es el gran número de evaluaciones de las derivadas necesarias por paso. El método de Runge-Kutta-Merson emplea cinco de tales evaluaciones, en tanto que en un método predictor-corrector se encuentra, con frecuencia, que bastan dos o tres correcciones. Este ahorro de tiempo en la computadora en el caso de métodos multipaso puede ser muy significativo en el caso de que en la función derivada se emplea el cálculo de funciones especiales o cuando la evalucación es complicada.

7.9 Programas desarrollados en Matlab

Esta sección proporciona los códigos de los programas desarrollados en Matlab para todos los ejercicios propuestos. A continuación se da una lista de todos ellos:

7.9.1. Regla trapezoidal
7.9.2. Método de Euler
7.9.3. Método de Runge-Kutta de segundo orden
7.9.4. Método de Runge-Kutta de tercer orden
7.9.5. Método de Runge-Kutta de cuarto orden
7.9.6. Método explícito para M = 1
7.9.7. Método explícito para M = 2
7.9.8. Método explícito para M = 3
7.9.9. Método multipaso lineal
7.9.10. Método de Euler para sistemas de ecuaciones
7.9.11. Método de Euler trapezoidal para sistemas de ecuaciones
7.9.12. Método de Runge-Kutta de segundo orden
7.9.13. Método de Runge-Kutta de cuarto orden

7.9.1 Regla trapezoidal

El método de la regla trapezoidal para la solución de una ecuación diferencial ordinaria se conoce como *método de Euler-Cauchy*. Se basa en la aproximación simple de una derivada por diferencias hacia delante. En Matlab, cuando se usa el signo % significa que se está haciendo un comentario. Adicionalmente las funciones propias de Matlab aparecen sombreadas.

Programa principal de la regla trapezoidal

```
% Regla trapezoidal para la solución de una ecuación diferencial ordinaria. El
% método indica que las variables bajo el signo de diferenciación se reemplazan por
% diferencias y las demás variables por valor promedio.
```

www.FreeLibros.me

```
% En este caso específico se programa la solución de la ecuación diferencial
% v' + 4v = exp(-2t).
% Aplicando el método se tiene:
% (v(n+1) - v(n))/Dt + 4((v(n+1) + v(n))/2) = (exp(-2*t(n+1))+exp(-2*t(n)))/2.
% De la ecuación anterior se despeja v(n+1) en función de v(n) y de la fuente.
clear all
clc
Tobs = 1;                          % Tiempo de observación.
h = 0.01;                          % Paso de integración.
t = 0:h:Tobs;                      % Vector de tiempos de integración.
N = length(t);                     % Número de muestras que se simulan.
v(1) = 1;                          % Condición inicial.
c1   = 2*(1/h + 2);                % Constante que acompaña a la fuente.
c2   = (1/h - 2)/(1/h + 2);        % Constante que acompaña a v(n).
% Ciclo iterativo para calcular la EDO para toda t.
for k = 2:N;
    v(k) = (exp(-2*t(k)) + exp(-2*t(k-1)))/c1 + c2*v(k-1);
end
% Solución analítica que sirve como referencia.
v1 = 0.5*exp(-2*t)+0.5*exp(-4*t);
% Grafica la solución analítica y la solución numérica.
plot(t,v,'or',t,v1,'b')
legend('Solución numérica','Solución analítica')
xlabel('Tiempo en segundos')
ylabel('Solución en p.u.')
```

7.9.2. Método de Euler

El método de Euler se obtiene reescribiendo la derivada de primer orden por la aproximación en diferencias hacia delante.

Programa principal del método de Euler

```
% Método de Euler para la solución de una ecuación diferencial ordinaria; el método
% se basa en la definición numérica de una derivada de primer orden.
% En este caso específico se programa la solución de la ecuación diferencial y' + 20y =
% 7*exp(-0.5t)
% Aplicando el método se tiene:    y(n+1) = y(n) + h*f(y(n),t(n))
clear all
clc
Tobs = 0.5;        % Tiempo de observación.
h = 0.001;         % Paso de integración.
t = 0:h:Tobs;      % Vector de tiempos de integración.
N = length(t);     % Número de muestras que se simulan.
y(1) = 5;          % Condición inicial.
% Ciclo iterativo para calcular la EDO para toda t.
for k=2:N
    y(k) = y(k-1) + h*(-20*y(k-1) + 7*exp(-0.5*t(k-1)));
end
% Solución analítica.
y1 = (181/39).*exp(-20*t) + (14/39).*exp(-0.5*t);
% Grafica la solución analítica y la solución numérica.
plot(t(1:10:N),y(1:10:N),'or',t,y1,'b')
legend('Solución numérica','Solución analítica')
xlabel('Tiempo en segundos')
ylabel('Solución en p.u.')
```

www.FreeLibros.me

7.9.3 Método de Runge-Kutta de segundo orden

El método de Runge-Kutta de segundo orden se deduce de la solución de la ecuación 7.13 con P = 2. En Matlab, cuando se usa el signo % significa que se está haciendo un comentario. Adicionalmente las funciones propias de Matlab aparecen sombreadas.

Programa principal del método de Runge-Kutta de segundo orden

```
% Método de Euler modificado (Runge-Kutta segundo orden) para la solución de una
% ecuación diferencial ordinaria; el método se basa en la expansión de Taylor de
% varias variables.
% En este caso específico se programa la solución de la ecuación diferencial
% i' + 0.4i = 0.2
clear all
clc
Tobs = 14;           % Tiempo de observación.
h = 0.1;             % Paso de integración.
t = 0:h:Tobs;        % Vector de tiempos de integración.
N = length(t);       % Número de muestras que se simulan.
i(1) = 0;            % Condición inicial.
k1(1) = 0;           % Inicializa k1.
k2(1) = 0;           % Inicializa k2.
% Ciclo iterativo para calcular la EDO para toda t.
for k=2:N
    k1(k) = h*(0.2 - 0.4*i(k-1));
    k2(k) = h*(0.2 - 0.4*(i(k-1)+k1(k)));
    i(k) = i(k-1)+(1/2)*(k1(k)+k2(k));
end
% Solución analítica.
i1 = 1/2 - (1/2)*exp(-0.4*t);
% Grafica la solución analítica y la solución numérica.
plot(t(1:5:N),i(1:5:N),'or',t,i1,'b')
legend('Solución numérica','Solución analítica')
xlabel('Tiempo en segundos')
ylabel('Solución en p.u.')
```

7.9.4 Método de Runge-Kutta de tercer orden

El método de Runge-Kutta de tercer orden se deduce de la solución de la ecuación 7.13 con P = 4 y r = 3. En Matlab, cuando se usa el signo % significa que se está haciendo un comentario. Adicionalmente las funciones propias de Matlab aparecen sombreadas.

Programa principal del método de Runge-Kutta de tercer orden

```
% Método de Runge-Kutta tercer orden para la solución de una ecuación diferencial
% ordinaria; el método se basa en la expansión de Taylor de varias variables.
% En este caso específico se programa la solución de la ecuación diferencial.
% w' + 6w = 8*t*exp(-t)
clear all
clc
Tobs = 10;          % Tiempo de observación.
h = 0.01;           % Paso de integración.
t = 0:h:Tobs;       % Vector de tiempos de integración.
N = length(t);      % Número de muestras que se simulan.
w(1) = 0;           % Condición inicial.
k1(1) = 0;          % Inicializa k1.
k2(1) = 0;          % Inicializa k2.
k3(1) = 0;          % Inicializa k3.
% Ciclo iterativo para calcular la EDO para toda t.
```

www.FreeLibros.me

```
for k=2:N
    k1(k) = h*(8*t(k-1)*exp(-t(k-1)) - 6*w(k-1));
    k2(k) = h*(8*(t(k-1)+h/2)*exp(-(t(k-1)+h/2)) - 6*(w(k-1)+(1/2)*h*k1(k)));
    k3(k) = h*(8*(t(k-1)+h)*exp(-(t(k-1)+h)) - 6*(w(k-1)+h*k1(k)));
    w(k) = w(k-1)+(1/6)*(k1(k)+4*k2(k)+k3(k));
end
% Solución analítica.
w1 = (8/25).*exp(-6*t) + (8/5).*t.*exp(-t) - (8/25).*exp(-t);
% Grafica la solución analítica y la solución numérica.
plot(t(1:20:N),w(1:20:N),'or',t,w1,'b')
legend('Solución numérica','Solución analítica')
xlabel('Tiempo en segundos')
ylabel('Solución en p.u.')
```

7.9.5 Método de Runge-Kutta de cuarto orden

El método de Runge-Kutta de cuarto orden se deduce de la solución de la ecuación 7.13 con P = 4 y r = 4. En Matlab, cuando se usa el signo % significa que se está haciendo un comentario. Adicionalmente las funciones propias de Matlab aparecen sombreadas.

Programa principal del método de Runge-Kutta de cuarto orden

```
% Método de Runge-Kutta cuarto orden para la solución de una ecuación diferencial
% ordinaria; el método se basa en la expansión de Taylor de varias variables.
% En este caso específico se programa la solución de la ecuación diferencial
%     y' + 2y = 0   con   y(0)=1
clear all
clc
Tobs = 5;           % Tiempo de observación.
h = 0.01;           % Paso de integración.
t = 0:h:Tobs;       % Vector de tiempos de integración.
N = length(t);      % Número de muestras que se simulan.
y(1) = 1;           % Condición inicial.
k1(1) = 0;          % Inicializa k1.
k2(1) = 0;          % Inicializa k2.
k3(1) = 0;          % Inicializa k3.
k4(1) = 0;          % Inicializa k4.
% Ciclo iterativo para calcular la EDO para toda t.
for k=2:N
    k1(k) = h*(- 2*y(k-1));
    k2(k) = h*(- 2*(y(k-1)+(1/2)*h*k1(k)));
    k3(k) = h*(- 2*(y(k-1)+(1/2)*h*k2(k)));
    k4(k) = h*(- 2*(y(k-1)+h*k3(k)));
    y(k) = y(k-1)+(1/6)*(k1(k)+2*k2(k)+2*k3(k)+k4(k));
end
% Solución analítica
y1 = 1.*exp(-2*t);
% Grafica la solución analítica y la solución numérica.
plot(t(1:10:N),y(1:10:N),'or',t,y1,'b')
legend('Solución numérica','Solución analítica')
xlabel('Tiempo en segundos')
ylabel('Solución en p.u.')
```

7.9.6 Método explícito para m = 1

Los métodos explícitos utilizan un polinomio interpolador para aproximar la función dentro del intervalo. Para el caso de m = 1, se obtiene un método de segundo orden. En Matlab, cuando se usa el signo % significa que se está haciendo un comentario. Adicionalmente, las funciones propias de Matlab aparecen sombreadas.

www.FreeLibros.me

Programa principal del método explícito para m = 1

```
% El método explícito con m = 1 para la solución de una ecuación diferencial
% ordinaria. En este caso específico se programa la solución de la ecuación
% diferencial   y' + 20y = 7*exp(-0.5t). Aplicando el método se tiene:
% y(n+1) = y(n) + (h/2)*(3*f(n) - f(n-1))
clear all
clc
Tobs = 0.5;             % Tiempo de observación.
h = 0.001;              % Paso de integración.
t = 0:h:Tobs;           % Vector de tiempos de integración.
N = length(t);          % Número de muestras que se simulan.
y(1) = 5;               % Primera condición de arranque.
y(2) = 2292/467;        % Segunda condición de arranque.
% Ciclo iterativo para calcular la EDO para toda t.
for k=3:N
    fn  = -20*y(k-1)+7*exp(-0.5*t(k-1));    % Función evaluada en k-1.
    fn1 = -20*y(k-2)+7*exp(-0.5*t(k-2));    % Función evaluada en k-2.
    y(k) = y(k-1) + (h/2)*(3*fn - fn1);     % Fórmula explícita para m=1.
end
% Solución analítica.
y1 = (181/39).*exp(-20*t) + (14/39).*exp(-0.5*t);
% Grafica la solución analítica y la solución numérica.
plot(t(1:10:N),y(1:10:N),'or',t,y1,'b')
legend('Solución numérica','Solución analítica')
xlabel('Tiempo en segundos')
ylabel('Solución en p.u.')
```

7.9.7 Método explícito para m = 2

Los métodos explícitos utilizan un polinomio interpolador para aproximar la función dentro del intervalo. Para el caso de m = 2 se obtiene un método de tercer orden.

Programa principal del método explícito para m = 2

```
% El método explicito con m = 2  para la solución de una ecuación diferencial
% ordinaria. En este caso específico se programa la solución de la ecuación
% diferencial.
%     y' + 20y = 7*exp(-0.5t)
% Aplicando el método se tiene:
% y(n+1) = y(n) + (h/12)*(23*f(n) - 16*f(n-1) + 5*f(n-2))
clear all
clc
Tobs = 0.5;             % Tiempo de observación.
h = 0.001;              % Paso de integración.
t = 0:h:Tobs;           % Vector de tiempos de integración.
N = length(t);          % Número de muestras que se simulan.
y(1) = 5;               % Primera condición de inicio.
y(2) = 2292/467;        % Segunda condición de inicio.
y(3) = 1691/351;        % Tercera condición de inicio.
% Ciclo iterativo para calcular la EDO para toda t.
for k=4:N
    fn  = -20*y(k-1)+7*exp(-0.5*t(k-1));              % Función evaluada en k-1.
    fn1 = -20*y(k-2)+7*exp(-0.5*t(k-2));              % Función evaluada en k-2.
    fn2 = -20*y(k-3)+7*exp(-0.5*t(k-3));              % Función evaluada en k-3.
    y(k) = y(k-1) + (h/12)*(23*fn - 16*fn1 + 5*fn2);  % Fórmula explícita para m = 2.
end
% Solución analítica.
y1 = (181/39).*exp(-20*t) + (14/39).*exp(-0.5*t);
% Grafica la solución analítica y la solución numérica.
```

www.FreeLibros.me

```
plot(t(1:10:N),y(1:10:N),'or',t,y1,'b')
legend('Solución numérica','Solución analítica')
xlabel('Tiempo en segundos')
ylabel('Solución en p.u.')
```

7.9.8 Método explícito para m = 3

Los métodos explícitos utilizan un polinomio interpolador para aproximar la función dentro del intervalo. Para el caso de m=3 se obtiene un método de cuarto orden.

Programa principal del método explícito para m = 3

```
% El método explícito con m = 3 para la solución de una ecuación diferencial
% ordinaria. En este caso específico se programa la solución de la ecuación
% diferencial   y' + 20y = 7*exp(-0.5t)
% Aplicando el método se tiene:
% y(n+1) = y(n) + (h/24)*(55*f(n) - 59*f(n-1) + 37*f(n-2) - 9*f(n-3))
clear all
clc
Tobs = 0.5;             % Tiempo de observación.
h = 0.001;              % Paso de integración.
t = 0:h:Tobs;           % Vector de tiempos de integración.
N = length(t);          % Número de muestras que se simulan.
y(1) = 5;               % Primera condición de arranque.
y(2) = 2292/467;        % Segunda condición de arranque.
y(3) = 1691/351;        % Tercera condición de arranque.
y(4) = 4261/901;        % Cuarta condición de arranque.
% Ciclo iterativo para calcular la EDO para toda t.
for k=5:N
   fn0 = -20*y(k-1)+7*exp(-0.5*t(k-1));               % Función evaluada en k-1.
   fn1 = -20*y(k-2)+7*exp(-0.5*t(k-2));               % Función evaluada en k-2.
   fn2 = -20*y(k-3)+7*exp(-0.5*t(k-3));               % Función evaluada en k-3.
   fn3 = -20*y(k-4)+7*exp(-0.5*t(k-4));               % Función evaluada en k-4.
   y(k) = y(k-1)+(h/24)*(55*fn0-59*fn1+37*fn2-9*fn3); % Fórmula explícita para m = 3.
end
% Solución analítica.
y1 = (181/39).*exp(-20*t) + (14/39).*exp(-0.5*t);
% Grafica la solución analítica y la solución numérica.
plot(t(1:5:N),y(1:5:N),'or',t,y1,'b')
legend('Solución numérica','Solución analítica')
xlabel('Tiempo en segundos')
ylabel('Solución en p.u.')
```

7.9.9 Método multipaso lineal

Los métodos multipaso lineales aproximan la primera derivada con dos, tres, cuatro o *n* puntos; el método resultante depende de esta aproximación.

Programa principal del método multipaso lineal

```
% Método multipaso lineal para la solución de una ecuación diferencial ordinaria. En
% este caso específico se programa la solución de la ecuación diferencial  y' + 20y =
% 7*exp(-0.5t)
% Aplicando uno de los métodos multipaso lineales se tiene:
% y(n+1) = (3/2)*f(n) - 3*f(n-1) + (5/2)*f(n-2) + 3*h*f(t(n-1),y(n-1))
clear all
clc
Tobs = 0.5;               % Tiempo de observación.
h = 0.001;                % Paso de integración.
```

www.FreeLibros.me

```
t = 0:h:Tobs;             % Vector de tiempos de integración.
N = length(t);            % Número de muestras que se simulan.
y(1) = 5;                 % Primera condición de inicio.
y(2) = 2292/467;          % Segunda condición de inicio.
% Ciclo iterativo para calcular la EDO para toda t.
for k=3:N
    fn1 = -20*y(k-2)+7*exp(-0.5*t(k-2));             % Función evaluada en k-2.
    fn2 = -20*y(k-1)+7*exp(-0.5*t(k-1));             % Función evaluada en k-1.
    y(k) = -4*y(k-1) + 5*y(k-2) + h*(4*fn2 + 2*fn1); % Fórmula del método multipaso
                                                     % lineal.
end
% Solución analítica.
y1 = (181/39).*exp(-20*t) + (14/39).*exp(-0.5*t);
% Grafica la solución analítica y la solución numérica.
plot(t(1:5:N),y(1:5:N),'or',t,y1,'b')
legend('Solución numérica','Solución analítica')
xlabel('Tiempo en segundos')
ylabel('Solución en p.u.')
```

7.9.10 Método de Euler para sistemas de ecuaciones

El método de Euler es una extensión del caso de una ecuación diferencial ordinaria a un sistema de ecuaciones diferenciales ordinarias.

Programa principal del método de Euler

```
% El método de Euler aplicado a un sistema de ecuaciones diferenciales ordinarias. En
% este caso específico se programa la solución de la ecuación diferencial    v'' +
% 3v' + 8v = 0
% Se hace la transformación   v' = z   y   z' = -3z -8v
% Condiciones iniciales   v(0) = 1   y   z(0) = 0
clear all
clc
Tobs = 5;           % Tiempo de observación.
h = 0.01;           % Paso de integración.
t = 0:h:Tobs;       % Vector de tiempos de integración.
N = length(t);      % Número de muestras que se simulan.
% Solución por el método de Euler.
v(1) = 1;    %  Condición inicial de v.
z(1) = 0;    %  Condición inicial de z.
% Ciclo iterativo para calcular la EDO para toda t.
for k=2:N
    v(k) = v(k-1) + h*(z(k-1));
    z(k) = z(k-1) + h*(-3*z(k-1)-8*v(k-1));
end
r = [1 3 8];                 % Coeficientes del polinomio característico de la EDO.
a = roots(r);                % Raíces del polinomio característico.
alf = real(a(1));            % Parte real de las raíces.
bet = imag(a(1));            % Parte imaginaria de las raíces.
v1 = exp(alf*t).*(cos(bet*t)+(-alf/bet)*sin(bet*t));   % Solución analítica.
% Grafica la solución analítica y la solución numérica.
figure(1),plot(t,v1,'r',t(1:10:N),v(1:10:N),'o')
legend('Solución analítica','Solución numérica de Euler')
xlabel('Tiempo en segundos')
ylabel('Solución en p.u.')
```

7.9.11 Método de Euler trapezoidal para sistemas de ecuaciones

El método de Euler trapezoidal es una modificación del esquema numérico de Euler que conduce a un esquema más preciso y más estable.

www.FreeLibros.me

Programa principal del método de Euler trapezoidal

```
% El método de Euler trapezoidal aplicado a un sistema de ecuaciones diferenciales
% ordinarias. En este caso específico se programa la solución de la ecuación
% diferencial   v" + 3v' + 8v = 0.
% Se hace la transformación    v' = z   y   z' = -3z -8v
% Condiciones iniciales      v(0) = 1   y   z(0) = 0
clear all
clc
Tobs = 5;           % Tiempo de observación.
h = 0.01;           % Paso de integración.
t = 0:h:Tobs;       % Vector de tiempos de integración.
N = length(t);      % Número de muestras que se simulan.
% Solución por el método de Euler trapezoidal.
v(1) = 1;           % Condición inicial de v.
z(1) = 0;           % Condición inicial de z.
% Ciclo iterativo para calcular la solución para todo el tiempo de observación.
for k=2:N
    tol = 1e-3;        % Tolerancia utilizada para la convergencia de cada iteración.
    v(k) = v(k-1);     % Valor inicial de v para entrar al ciclo iterativo.
    z(k) = z(k-1);     % Valor inicial de z para entrar al ciclo iterativo.
    a = v(k)+1; b = z(k)+1;   % Se asignan variables para verificar la convergencia.
    % Ciclo que verifica la convergencia comparando los valores nuevos con los
    % anteriores.
    while abs(a-v(k)) > tol & abs(b-z(k))>tol
     a = v(k);                                                   % Valor anterior de v.
     b = z(k);                                                   % Valor anterior de z.
     v(k)=v(k-1)+(h/2)*(z(k-1)+z(k));                            % Valor nuevo de v.
     z(k)=z(k-1)+(h/2)*((-3*z(k-1)-8*v(k-1))+(-3*z(k)-8*v(k-1))); % Valor nuevo de z.
    end
end
% Solución analítica.
r = [1 3 8];                  % Coeficientes del polinomio característico de la EDO.
a = roots(r);                 % Raíces del polinomio característico.
alf = real(a(1));             % Parte real de las raíces.
bet = imag(a(1));             % Parte imaginaria de las raíces.
v1 = exp(alf*t).*(cos(bet*t)+(-alf/bet)*sin(bet*t));   % Solución analítica.
% Grafica la solución analítica y la solución numérica.
figure(1),plot(t,v1,'r',t(1:10:N),v(1:10:N),'o')
legend('Solución analítica','Solución numérica de Euler trapezoidal')
xlabel('Tiempo en segundos')
ylabel('Solución en p.u.')
```

7.9.12 Método de Runge-Kutta de segundo orden

El método de Runge-Kutta para sistemas de ecuación es una simple expansión del método para una sola ecuación.

Programa principal del método de Runge-Kutta de segundo orden

```
% El método de Runge-Kutta de segundo orden aplicado a un sistema de ecuaciones
% diferenciales ordinarias. En este caso específico se programa la solución del
% sistema de ecuaciones X' + AX = B.
%   A = [ 3 2       B = [ exp(-2t)          X = [ i
%         2 8 ]           exp(-3t) ]              v ]
% Condiciones iniciales    i(0) = 0   y   v(0) = 0
clear all
clc
Tobs = 5;                % Tiempo de observación.
h = 0.001;               % Paso de integración.
t = 0:h:Tobs;            % Vector de tiempos de integración.
```

www.FreeLibros.me

```
Ns = length(t);          % Número de muestras que se simulan.
A = -[3 2; 2 8];         % Matriz de coeficientes.
i(1) = 0;                % Condición inicial de i.
v(1) = 0;                % Condición inicial de v.
k1(1) = 0;               % Inicializa k1.
k2(1) = 0;               % Inicializa k2.
m1(1) = 0;               % Inicializa m1.
m2(1) = 0;               % Inicializa m2.
% Ciclo iterativo para resolver el sistema de ecuaciones diferenciales para toda t.
for k=2:Ns
    k1(k) = h*(-3*i(k-1)-2*v(k-1)+exp(-2*t(k-1)));                % Cálculo de k1.
    m1(k) = h*(-2*i(k-1)-8*v(k-1)+exp(-3*t(k-1)));                % Cálculo de m1.
    k2(k) = h*(-3*(i(k-1)+k1(k))-2*(v(k-1)+m1(k))+exp(-2*t(k))); % Cálculo de k2.
    m2(k) = h*(-2*(i(k-1)+k1(k))-8*(v(k-1)+m1(k))+exp(-3*t(k))); % Cálculo de m2.
    i(k) = i(k-1)+(1/2)*(k1(k)+k2(k));                            % Cálculo de la
                                                                  % nueva i.
    v(k) = v(k-1)+(1/2)*(m1(k)+m2(k));                            % Cálculo de la
                                                                  % nueva v.
end
% Solución analítica para efectos de comparación.
[M,L]=eig(A);   % Cálculo de valores y vectores propios.
N = M.';        % Cálculo de la inversa de M, en este caso igual a su transpuesta.
% Asignación de vectores izquierdos y derechos a diferentes variables.
M11=M(1,1); M12=M(1,2); M21=M(2,1); M22=M(2,2);
N11=N(1,1); N12=N(1,2); N21=N(2,1); N22=N(2,2);
L1=L(1,1);  L2=L(2,2);   % Asignación de los valores propios a diferentes variables.
E1=-2;      E2=-3;       % Asignación de los coeficientes de amortiguamiento.
% Convoluciones.
L1E1 = (1/(L1-E1))*(exp(L1.*t)-exp(E1.*t));
L2E1 = (1/(L2-E1))*(exp(L2.*t)-exp(E1.*t));
L1E2 = (1/(L1-E2))*(exp(L1.*t)-exp(E2.*t));
L2E2 = (1/(L2-E2))*(exp(L2.*t)-exp(E2.*t));
it = M11*N11*L1E1 + M12*N21*L2E1 + M11*N12*L1E2 + M12*N22*L2E2;    % Solución para i.
vt = M21*N11*L1E1 + M22*N21*L2E1 + M21*N12*L1E2 + M22*N22*L2E2;    % Solución para v.
% Grafica la solución analítica y la solución numérica.
figure(1),plot(t(1:100:Ns),i(1:100:Ns),'or',t,it,'--')
legend('Solución numérica de la variable in','Solución analítica')
xlabel('Tiempo en segundos')
ylabel('Solución en p.u.')
% Grafica la solución analítica y la solución numérica.
figure(2),plot(t(1:100:Ns),v(1:100:Ns),'or',t,vt,'--b')
legend('Solución numérica de la variable vn','Solución analítica')
xlabel('Tiempo en segundos')
ylabel('Solución en p.u.')
% Grafica la solución analítica y la solución numérica.
figure(3),plot(t(1:100:Ns),i(1:100:Ns),'or',t(1:100:Ns),v(1:100:Ns),'ob',t,it,'r',t,v
t,'b')
legend('Solución numérica de la variable In','Solución numérica de la variable
Vn',...
       'Solución analítica de la variable In','Solución analítica de la variable Vn')
xlabel('Tiempo en segundos')
ylabel('Solución en p.u.')
```

7.9.13 Método de Runge-Kutta de cuarto orden

El método de Runge-Kutta para sistemas de ecuación es una simple expansión del método para una sola ecuación.

Programa principal del método de Runge-Kutta de cuarto orden

```
% El método de Runge-Kutta de cuarto orden aplicado a un sistema de ecuaciones
% diferenciales ordinarias. En este caso específico se programa la solución del
% sistema de ecuaciones X' + AX = B.
```

www.FreeLibros.me

```
%   A = [ 3 2        B = [ exp(-2t)        X = [ i
%         2 8 ]            exp(-3t) ]            v ]
% Condiciones iniciales    i(0) = 0   y   v(0) = 0
clear all
clc
Tobs = 5;            % Tiempo de observación.
h = 0.001;           % Paso de integración.
t = 0:h:Tobs;        % Vector de tiempos de integración.
Ns = length(t);      % Número de muestras que se simulan.
A = -[3 2; 2 8];     % Matriz de coeficientes.
i(1) = 0;            % Condición inicial de i.
v(1) = 0;            % Condición inicial de v.
k1(1) = 0;           % Inicializa k1.
k2(1) = 0;           % Inicializa k2.
k3(1) = 0;           % Inicializa k3.
k4(1) = 0;           % Inicializa k4.
m1(1) = 0;           % Inicializa m1.
m2(1) = 0;           % Inicializa m2.
m3(1) = 0;           % Inicializa m3.
m4(1) = 0;           % Inicializa m4.
% Ciclo iterativo para resolver el sistema de ecuaciones diferenciales para toda t.
for k=2:Ns
    k1(k) = h*(-3*i(k-1)-2*v(k-1)+exp(-2*t(k-1)));       % Cálculo de k1.
    m1(k) = h*(-2*i(k-1)-8*v(k-1)+exp(-3*t(k-1)));       % Cálculo de m1.
    k2(k) = h*(-3*(i(k-1)+k1(k)/2)-2*(v(k-1)+m1(k)/2)
    +exp(-2*(t(k)+h/2)));                                % Cálculo de k2.
    m2(k) = h*(-2*(i(k-1)+k1(k)/2)-8*(v(k-1)+m1(k)/2)
    +exp(-3*(t(k)+h/2)));                                % Cálculo de m2.
    k3(k) = h*(-3*(i(k-1)+k2(k)/2)-2*(v(k-1)+m2(k)/2)
    +exp(-2*(t(k)+h/2)));                                % Cálculo de k3.
    m3(k) = h*(-2*(i(k-1)+k2(k)/2)-8*(v(k-1)+m2(k)/2)
    +exp(-3*(t(k)+h/2)));                                % Cálculo de m3.
    k4(k) = h*(-3*(i(k-1)+k3(k))-2*(v(k-1)+m2(k))
    +exp(-2*(t(k)+h)));                                  % Cálculo de k4.
    m4(k) = h*(-2*(i(k-1)+k2(k))-8*(v(k-1)+m2(k))
    +exp(-3*(t(k)+h)));                                  % Cálculo de m4.
    i(k) = i(k-1)+(1/6)*(k1(k)+2*k2(k)+2*k3(k)+k4(k));   % Cálculo de la nueva i.
    v(k) = v(k-1)+(1/6)*(m1(k)+2*m2(k)+2*m3(k)+m4(k));   % Cálculo de la nueva v.
end
% Solución analítica para efectos de comparación.
[M,L]=eig(A);   % Cálculo de valores y vectores propios.
N = M.';        % Cálculo de la inversa de M, en este caso igual a su transpuesta.
% Asignación de vectores izquierdos y derechos a diferentes variables.
M11=M(1,1); M12=M(1,2); M21=M(2,1); M22=M(2,2);
N11=N(1,1); N12=N(1,2); N21=N(2,1); N22=N(2,2);
L1=L(1,1);  L2=L(2,2);   % Asignación de los valores propios a diferentes variables.
E1=-2;      E2=-3;       % Asignación de los coeficientes de amortiguamiento.
% Convoluciones
L1E1 = (1/(L1-E1))*(exp(L1.*t)-exp(E1.*t));
L2E1 = (1/(L2-E1))*(exp(L2.*t)-exp(E1.*t));
L1E2 = (1/(L1-E2))*(exp(L1.*t)-exp(E2.*t));
L2E2 = (1/(L2-E2))*(exp(L2.*t)-exp(E2.*t));
it = M11*N11*L1E1 + M12*N21*L2E1 + M11*N12*L1E2 + M12*N22*L2E2;   % Solución para i.
vt = M21*N11*L1E1 + M22*N21*L2E1 + M21*N12*L1E2 + M22*N22*L2E2;   % Solución para v.
% Grafica la solución analítica y la solución numérica.
figure(1),plot(t(1:100:Ns), i(1:100:Ns),'or',t,it,'--')
legend('Solución numérica de la variable in','Solución analítica')
xlabel('Tiempo en segundos')
ylabel('Solución en p.u.')
% Grafica la solución analítica y la solución numérica.
figure(2),plot(t(1:100:Ns),v(1:100:Ns),'or',t,vt,'--b')
legend('Solución numérica de la variable vn','Solución analítica')
xlabel('Tiempo en segundos')
ylabel('Solución en p.u.')
% Grafica la solución analítica y la solución numérica.
```

www.FreeLibros.me

```
figure(3),plot(t(1:100:Ns),i(1:100:Ns),'or',t(1:100:Ns),v(1:100:Ns),'ob',t,it,'r',t,
vt,'b')
legend('Solución numérica de la variable In','Solución numérica de la variable
Vn',...'Solución analítica de la variable In','Solución analítica de la variable Vn')
xlabel('Tiempo en segundos')
ylabel('Solución en p.u.')
```

Problemas propuestos

7.10.1 Utilizando el método de la serie de Taylor, resuelva la ecuación

$$y' = \frac{3}{2+y}$$

con las condiciones iniciales $y(0)=1$ y $p=3$.

7.10.2 Implemente la regla trapezoidal para resolver la ecuación

$$\frac{dx}{dt} + 2x = 5e^{-7t}$$

con las condiciones iniciales $x(0)=1$ y $\Delta t = 0.01,\ T_{\text{máx}} = 5$.

7.10.3 Implemente la regla trapezoidal para resolver la ecuación

$$\frac{dy}{dt} + 3y = 4te^{-t}$$

con las condiciones iniciales $y(0)=0.1$ y $\Delta t = 0.005,\ T_{\text{máx}} = 10$.

7.10.4 Implemente la regla trapezoidal para resolver la ecuación

$$\frac{dy}{dt} + 3.2y = 7$$

con las condiciones iniciales $y(0)=0.3$ y $\Delta t = 0.05,\ T_{\text{máx}} = 2$.

7.10.5 Implemente el método de Euler para resolver la ecuación

$$\frac{da}{dt} + 2a = 2e^{-0.2t}$$

con las condiciones iniciales $a(0)=0.5$ y $\Delta t = 0.01,\ T_{\text{máx}} = 1$.

7.10.6 Implemente el método de Euler para resolver la ecuación

$$\frac{dw}{dt} + 7w = e^{-7t}$$

con las condiciones iniciales $w(0)=0.01$ y $\Delta t = 0.001,\ T_{\text{máx}} = 1$.

7.10.7 Implemente el método de Euler para resolver la ecuación

$$\frac{dw}{dt} + 8.5w = 3t$$

con las condiciones iniciales $w(0)=1$ y $\Delta t = 0.001,\ T_{\text{máx}} = 1$.

www.FreeLibros.me

7.10.8 Por el método de Runge-Kutta de segundo orden, resuelva la ecuación

$$\frac{dv}{dt} + 3v = 16$$

con las condiciones iniciales $v(0) = 1$ y $\Delta t = 0.1$, $T_{\text{máx}} = 2$.

7.10.9 Por el método de Runge-Kutta de segundo orden, resuelva la ecuación

$$\frac{di}{dt} + 2i = 2\cos(377t)$$

con las condiciones iniciales $i(0) = 2$ y $\Delta t = 0.001$, $T_{\text{máx}} = 3$.

7.10.10 Por el método de Runge-Kutta de segundo orden, resuelva la ecuación

$$\frac{di}{dt} + 6.5i = e^{-9t}$$

con las condiciones iniciales $i(0) = 1$ y $\Delta t = 0.0001$, $T_{\text{máx}} = 1$.

7.10.11 Usando el método de Runge-Kutta de tercer orden, resuelva la ecuación

$$\frac{di}{dt} + \frac{1}{2}i = \operatorname{sen}(3t)$$

con las condiciones iniciales $i(0) = 0$ y $\Delta t = 0.1$, $T_{\text{máx}} = 3$.

7.10.12 Usando el método de Runge-Kutta de tercer orden, resuelva la ecuación

$$\frac{dz}{dt} + 9z = e^{-4t}\operatorname{sen}(500t)$$

con las condiciones iniciales $z(0) = 1$ y $\Delta t = 0.01$, $T_{\text{máx}} = 1$.

7.10.13 Usando el método de Runge-Kutta de tercer orden, resuelva la ecuación

$$\frac{dz}{dt} + 2.1z = 5e^{-t}$$

con las condiciones iniciales $z(0) = 2$ y $\Delta t = 0.01$, $T_{\text{máx}} = 7$.

7.10.14 Por el método de Runge-Kutta clásico (cuarto orden), resuelva la ecuación

$$\frac{dw}{dt} + 7w = 5\ln t$$

con las condiciones iniciales $w(0.1) = 1$ y $\Delta t = 0.0001$, $T_{\text{máx}} = 1$.

7.10.15 Por el método de Runge-Kutta clásico (cuarto orden), resuelva la ecuación

$$\frac{dh}{dt} + h = 3t^2e^{-4t}$$

con las condiciones iniciales $h(0) = 2$ y $\Delta t = 0.001$, $T_{\text{máx}} = 5$.

7.10.16 Por el método de Runge-Kutta clásico (cuarto orden), resuelva la ecuación

$$\frac{dv}{dt} + 1e^{6}v = \cos(377t)$$

www.FreeLibros.me

con las condiciones iniciales $v(0)=0$, $T_{\text{máx}}=0.017$ con $\Delta t_1=0.00001$ y $\Delta t_2=0.000001$.

7.10.17 Por el método explícito con m = 1, resuelva la ecuación

$$\frac{df}{dt}+4f=5t^3\operatorname{sen}(120\pi t)$$

con las condiciones iniciales $f(0)=0$, $f(\Delta t)=0$, $\Delta t=0.00005$ y $T_{\text{máx}}=0.2$.

7.10.18 Por el método explícito con m = 1, resuelva la ecuación

$$\frac{dg}{dt}+3g=2\cos(360\pi t)$$

con las condiciones iniciales $g(0)=1$, $g(\Delta t)=9\,958/9\,959$, $\Delta t=0.0001$ y $T_{\text{máx}}=2$.

7.10.19 Por el método explícito con m = 1, resuelva la ecuación

$$\frac{dz}{dt}+8.2z=10$$

con las condiciones iniciales $z(0)=0.2$, $z(\Delta t)=1\,156/5\,549$, $\Delta t=0.001$ y $T_{\text{máx}}=1$.

7.10.20 Por el método explícito con m = 2, resuelva la ecuación

$$\frac{dv}{dt}+2v=5$$

con las condiciones iniciales $v(0)=1$, $v(\Delta t)=104/101$, $v(2\Delta t)=8\,101/7\,651$, $\Delta t=0.01$ y $T_{\text{máx}}=5$.

7.10.21 Por el método explícito con m = 2, resuelva la ecuación

$$\frac{di}{dt}+5i=8$$

con las condiciones iniciales $i(0)=2$, $i(\Delta t)=1\,001/501$, $i(2\Delta t)=2\,006/1\,005$, $\Delta t=0.001$ y $T_{\text{máx}}=2$.

7.10.22 Por el método explícito con m = 2, resuelva la ecuación

$$\frac{dq}{dt}+3.6q=te^{-t}$$

con las condiciones iniciales $q(0)=1$, $q(\Delta t)=832/835$, $q(2\Delta t)=969/976$, $\Delta t=0.001$ y $T_{\text{máx}}=10$.

7.10.23 Por el método explícito con m = 3, resuelva la ecuación

$$\frac{dx}{dt}+8x=15$$

con las condiciones iniciales $x(0)=6$, $x(\Delta t)=1\,271/213$, $x(2\Delta t)=997/168$, $x(3\Delta t)=543/92$, $\Delta t=0.001$ y $T_{\text{máx}}=1$.

7.10.24 Por el método explícito con m = 3, resuelva la ecuación

$$\frac{dy}{dt}+3y=2e^{-2t}$$

con las condiciones iniciales $y(0)=10$, $y(\Delta t)=1\,867/192$, $y(2\Delta t)=747/79$, $y(3\Delta t)=6\,225/677$, $\Delta t=0.01$ y $T_{\text{máx}}=3$.

www.FreeLibros.me

7.10.25 Por el método explícito con m = 3, resuelva la ecuación

$$\frac{dz}{dt}+1.9y=te^{-0.01t}$$

con las condiciones iniciales $z(0)=1,\ z(\Delta t)=578/695,\ z(2\Delta t)=597/851,\ z(3\Delta t)=928/1539,\ \Delta t=0.1$ y $T_{\text{máx}}=1000$.

7.10.26 Implemente el método de Euler para resolver el siguiente sistema de ecuaciones

$$y_1'+2y_1+3y_2=6$$

$$y_2'+y_1+5y_2=2$$

con las condiciones iniciales $y_1(0)=1, y_2(0)=1,\ \Delta t=0.05$ y $T_{\text{máx}}=5$.

7.10.27 Implemente el método de Euler para resolver el siguiente sistema de ecuaciones

$$\frac{d}{dt}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}+\begin{bmatrix}9&3&1\\2&8&4\\3&2&6\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}=\begin{bmatrix}4\\7\\5\end{bmatrix}$$

con las condiciones iniciales $x_1(0)=1, x_2(0)=1, x_3(0)=1,\ \Delta t=0.01$ y $T_{\text{máx}}=1$.

7.10.28 Implemente el método de Euler para resolver el siguiente sistema de ecuaciones

$$\frac{d}{dt}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}+\begin{bmatrix}6&7&5\\3&9&1\\5&1&7\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}=\begin{bmatrix}1\\1\\1\end{bmatrix}$$

con las condiciones iniciales $x_1(0)=1, x_2(0)=0, x_3(0)=-1,\ \Delta t=0.001$ y $T_{\text{máx}}=15$.

7.10.29 Implemente el método de Euler trapezoidal para resolver el siguiente sistema de ecuaciones

$$\frac{dx_1}{dt}+3x_1+2x_2+x_3=2$$

$$\frac{dx_2}{dt}+x_1+7x_2+x_3=6$$

$$\frac{dx_3}{dt}+2x_1+8x_2+3x_3=7$$

con las condiciones iniciales $x_1(0)=0.5, x_2(0)=0.2, x_3(0)=0.7,\ \Delta t=0.01$ y $T_{\text{máx}}=4$.

7.10.30 Implemente el método de Euler trapezoidal para resolver el siguiente sistema de ecuaciones

$$\frac{d}{dt}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}+\begin{bmatrix}5&1&1\\1&3&2\\1&2&4\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}=\begin{bmatrix}1\\1\\1\end{bmatrix}$$

con las condiciones iniciales $x_1(0)=0, x_2(0)=0.1, x_3(0)=0.2,\ \Delta t=0.001$ y $T_{\text{máx}}=4$.

www.FreeLibros.me

7.10.31 Utilice el método de Euler trapezoidal para resolver el siguiente sistema de ecuaciones

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 6 & 2 & 3 \\ 1 & 8 & 7 \\ 3 & 4 & 9 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 6 \end{bmatrix}$$

con las condiciones iniciales $x_1(0) = 0.05$, $x_2(0) = 0.11$, $x_3(0) = 0$, $\Delta t = 0.001$ y $T_{\text{máx}} = 2.5$.

7.10.32 Utilice el método de Euler trapezoidal para resolver el siguiente sistema de ecuaciones

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 9 & 2 & 3 \\ 3 & 8 & 1 \\ 1 & 0 & 9 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 9 \\ 7 \\ 1 \end{bmatrix}$$

con las condiciones iniciales $x_1(0) = 1$, $x_2(0) = 0$, $x_3(0) = 0$, $\Delta t = 0.0001$ y $T_{\text{máx}} = 1.5$.

7.10.33 Implemente el método de Runge-Kutta de segundo orden para resolver el siguiente sistema de ecuaciones

$$\frac{dx_1}{dt} + 32x_1 + 12x_2 + 7x_3 = 2\,\text{sen}(377t)$$

$$\frac{dx_2}{dt} + 13x_1 + 42x_2 + 21x_3 = 6\cos(754t)$$

$$\frac{dx_3}{dt} + 9x_1 + 28x_2 + 63x_3 = 7te^{-t}$$

con las condiciones iniciales $x_1(0) = 0.01$, $x_2(0) = 0.02$, $x_3(0) = 0.01$, $\Delta t = 0.0001$ y $T_{\text{máx}} = 10$.

7.10.34 Implemente el método de Runge-Kutta de segundo orden para resolver el siguiente sistema de ecuaciones

$$\frac{dx_1}{dt} + 15x_1 + 2x_2 + 3x_3 = \tan(t)$$

$$\frac{dx_2}{dt} + 3x_1 + 12x_2 + 1x_3 = 6\cos(7t)$$

$$\frac{dx_3}{dt} + 2x_1 + 8x_2 + 23x_3 = 7t^3e^{-10t}$$

con las condiciones iniciales $x_1(0) = 1$, $x_2(0) = 1$, $x_3(0) = 1$, $\Delta t = 0.0001$ y $T_{\text{máx}} = \pi$.

7.10.35 Implemente el método de Runge-Kutta de segundo orden para resolver el siguiente sistema de ecuaciones

$$\frac{dx_2}{dt} + 32x_1 + 14x_2 + 19x_3 = 7$$

$$\frac{dx_2}{dt} + 7x_1 + 51x_2 + 25x_3 = \cos(t)$$

$$\frac{dx_3}{dt} + 11x_1 + 29x_2 + 47x_3 = \log_2(1+t)e^{-0.5t}$$

con las condiciones iniciales $x_1(0) = 0.1$, $x_2(0) = 0$, $x_3(0) = 0$, $\Delta t = 0.01$ y $T_{\text{máx}} = 25$.

www.FreeLibros.me

7.10.36 Implemente el método de Runge-Kutta de cuarto orden para resolver el siguiente sistema de ecuaciones

$$\frac{dx_1}{dt}+2x_1+0.1x_2+0.2x_3=\operatorname{sen}(240\pi t)$$

$$\frac{dx_2}{dt}+0.1x_1+4x_2+0.3x_3=\cos(540\pi t)$$

$$\frac{dx_3}{dt}+0.9x_1+0.8x_2+6x_3=17t^4e^{-5t}$$

con las condiciones iniciales $x_1(0) = 0$, $x_2(0) = 0$, $x_3(0) = 1$, $\Delta t = 0.0001$ y $T_{\text{máx}} = 4$.

7.10.37 Implemente el método de Runge-Kutta de cuarto orden para resolver el siguiente sistema de ecuaciones

$$\frac{dx_1}{dt}+22x_1+10x_2+4x_3=17$$

$$\frac{dx_2}{dt}+3x_1+32x_2+1x_3=64$$

$$\frac{dx_3}{dt}+2x_1+8x_2+25x_3=7$$

con las condiciones iniciales $x_1(0) = 0.1$, $x_2(0) = 0.2$, $x_3(0) = 0.1$, $\Delta t = 0.001$ y $T_{\text{máx}} = 0.5$.

7.10.38 Implemente el método de Runge-Kutta de cuarto orden para resolver el siguiente sistema de ecuaciones

$$\frac{dx_1}{dt}+0.06x_1+0.03x_2+0.01x_3=1$$

$$\frac{dx_2}{dt}+0.01x_1+0.09x_2+0.04x_3=0$$

$$\frac{dx_3}{dt}+0.02x_1+0.01x_2+0.08x_3=e^{-0.05t}$$

con las condiciones iniciales $x_1(0) = 0.01$, $x_2(0) = 0$, $x_3(0) = 0$, $\Delta t = 0.01$ y $T_{\text{máx}} = 150$.

7.10.39 Según la ley de enfriamiento de Newton, la velocidad a la que se enfría un objeto es proporcional a la diferencia de temperatura entre el objeto y el medio ambiente que rodea al objeto. La ecuación diferencial del proceso es

$$\frac{dT}{dt}=k(T-T_a),$$

donde T representa la temperatura del objeto en el tiempo t, T_a la temperatura del medio ambiente y k es la constante de proporcionalidad. Si la temperatura del medio ambiente es de 28° C, la temperatura inicial del objeto es de 100° C y $k=\frac{1}{12}\ln\left(\frac{13}{18}\right)$. Usando el método de Runge-Kutta de cuarto orden, determine la temperatura aproximada del cuerpo después de 12 minutos. Compare su resultado con el valor exacto de la solución.

www.FreeLibros.me

7.10.40 A principios del siglo xx, A. J. Lotka y V. Volterra desarrollaron en forma independiente un modelo matemático para describir dos especies en competencia, donde una de ellas es un depredador con población *y* y la otra es la presa con población *x*. Las ecuaciones que describen el proceso son

$$\frac{dx}{dt}=Ax-Bxy$$

$$\frac{dy}{dt}=-Cy+Dxy$$

donde $A,B,C,D>0$ representan las tasas de natalidad y mortalidad de la presa *x*, y las tasas de mortalidad y natalidad del depredador *y*, respectivamente. Si suponemos que $A=2,\ B=2,\ C=1,\ D=1$ y las poblaciones iniciales de presa y depredador son $x(0)=1,\ y(0)=3$, respectivamente. Use el método de Runge-Kutta de cuarto orden con un tamaño de paso de 0.125 para aproximarse a la solución en el intervalo de tiempo $0\le t\le 10$.

7.10.41 Un modelo para la difusión de una enfermedad en una población está dado por el sistema

$$\frac{dS}{dt}=-ASI$$

$$\frac{dI}{dt}=ASI-BI$$

donde A, B, >0 representan las tasas de transmisión y de recuperación de la enfermedad, respectivamente, y S e I la población susceptible de ser infectada y la población infectada en el tiempo t (días). Considere que $A = 0.005$, $B = 0.5$. Si la población inicial de susceptibles e infectados es de 700 y 1 respectivamente, determine el comportamiento de la enfermedad durante los primeros 20 días.

7.10.42 El movimiento de una luna que se encuentra en una órbita plana en torno de un planeta es descrito por el sistema

$$\frac{d^2x}{dt^2}=-g\frac{mx}{r^3}$$

$$\frac{d^2y}{dt^2}=-g\frac{my}{r^3}$$

donde $r=\sqrt{x^2+y^2}$, g es la constante gravitacional y m es la masa de la luna. A fin de simplificar, se supone que $gm = 1$. Si $x(0)=1,\ x'(0)=0,\ y(0)=0,\ y'(0)=1$, la órbita es un círculo de radio 1. Haciendo $x_1=x,\ x_2=x',\ x_3=y,\ x_4=y'$, reduzca el sistema a un sistema de cuatro ecuaciones de primer orden. Use un tamaño de paso $\frac{2\pi}{1024}$ y el método de Runge-Kutta de cuarto orden para obtener un periodo de la órbita lunar.

7.10.43 Si se drena el agua de un tanque cilíndrico vertical por medio de una válvula en la base del cilindro. La tasa de disminución del nivel del agua es

$$\frac{dy}{dt}=-k\sqrt{y},$$

donde k es una constante que depende de la válvula, y es la altura del agua dentro del cilindro medida en metros y t es el tiempo medido en minutos. Si $k = 0.05$ y la altura inicial es de 3 m, determine el tiempo necesario para vaciar el tanque.

www.FreeLibros.me

Capítulo 8

Valores y vectores propios

8.1 Introducción

Al tener una transformación lineal de la forma

$$\mathbf{Y} = \mathbf{AX} \tag{8.1}$$

donde,

$$\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & \vdots \\ \cdots & \cdots & \ddots & \vdots \\ a_{n1} & \cdots & \cdots & a_{nn} \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \tag{8.2a, b, c}$$

Uno de los problemas más trascendentes es determinar qué vectores, si los hay, no cambian de dirección. Como dos vectores no triviales tienen la misma dirección si y sólo si uno de ellos es múltiplo escalar diferente de cero del otro, esto equivale al problema de determinar aquellos vectores **X** cuyas imágenes **Y** son de la forma

$$\mathbf{Y} = \lambda \mathbf{X} \tag{8.3}$$

es decir, aquellos vectores **X** tales que se cumple que

$$\mathbf{AX} = \lambda \mathbf{X} \tag{8.4}$$

Reacomodando algebraicamente esta ecuación, se llega a

$$(\mathbf{A} - \lambda \mathbf{I})\mathbf{X} = \mathbf{0} \tag{8.5}$$

Corolario 8.1 Un sistema homogéneo $\mathbf{AX} = \mathbf{0}$ de n ecuaciones lineales con n incógnitas tiene una solución no trivial, es decir una solución distinta de $x_1 = x_2 = \cdots = x_n = 0$, si y sólo si el determinante de los coeficientes $|\mathbf{A}|$ es igual a 0. En forma más concreta, si la matriz de los coeficientes **A** de un sistema homogéneo de $n-1$ ecuaciones lineales con n incógnitas $\mathbf{AX} = \mathbf{0}$ tiene rango $n-1$, entonces una solución completa del sistema es

www.FreeLibros.me

$$x_1 = c|\mathbf{M}_1|,\ x_2 = -c|\mathbf{M}_2|, \ldots, x_n = (-1)^{n+1} c|\mathbf{M}_n| \tag{8.6}$$

donde $\mathbf{M}_j$ es la *submatriz* de $(n-1)\times(n-1)$ obtenida a partir de **A** eliminando su *j*-ésima columna. Esta condición lleva a

$$|\mathbf{A}-\lambda\mathbf{I}| = \begin{vmatrix} a_{11}-\lambda & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22}-\lambda & \cdots & \vdots \\ \cdots & \cdots & \ddots & \vdots \\ a_{n1} & \cdots & \cdots & a_{nn}-\lambda \end{vmatrix} = 0 \tag{8.7}$$

Como resultado se obtiene, obviamente, una ecuación de grado n en el parámetro λ con primer coeficiente $(-1)^n$, así se tiene que

$$|\mathbf{A}-\lambda\mathbf{I}| = (-1)^n\left[\lambda^n - \beta_1\lambda^{n-1} + \beta_2\lambda^{n-2} + \cdots - \beta_{n-1}\lambda + \beta_n\right] = 0 \tag{8.8}$$

Para valores de λ que satisfacen esta ecuación, y sólo para esos valores, la ecuación matricial $(\mathbf{A}-\lambda\mathbf{I})\mathbf{X} = \mathbf{0}$ (ecuación 8.5), tiene vectores solución no triviales. Las n raíces de la ecuación (8.8), se llaman *raíces características*, *valores característicos*, *valores propios*, *autovalores* o *eigenvalores* de la matriz **A** y las soluciones no triviales correspondientes de la ecuación (8.5) son los *vectores característicos*, *vectores propios*, *autovectores* o *eigenvectores* de **A**. De esta forma se está pasando del problema $\mathbf{Y} = \mathbf{AX}$ a un problema de determinación de *valores propios* y *vectores propios*.

Teorema 8.1

Si

$$\lambda^n - \beta_1\lambda^{n-1} + \beta_2\lambda^{n-2} + \cdots + (-1)^{n-1}\beta_{n-1}\lambda + (-1)^n\beta_n = 0 \tag{8.9}$$

es la ecuación característica de una matriz cuadrada **A**, entonces β_i es igual a la suma de todos los menores principales de orden i en **A**. Para $i = 1$, se tiene el caso especial de,

$$\beta_1 = \lambda_1 + \lambda_2 + \cdots + \lambda_n = a_{11} + a_{22} + \cdots + a_{nn} \tag{8.10a}$$

lo que se resume como sigue.

La suma de los n valores propios de **A** es igual a la suma de sus n entradas diagonales

$$\lambda_1 + \lambda_2 + \cdots + \lambda_n = a_{11} + a_{22} + \cdots + a_{nn} \tag{8.10b}$$

Esta suma se conoce como *traza de* **A**. Aún más, el producto de los n valores propios es igual al determinante de **A**. No hay duda que para muchas matrices, el problema de valores propios es más complejo que resolver el sistema de ecuaciones lineales. Sin embargo, para el caso de sistemas de ecuaciones diferenciales, conduce al proceso de diagonalización que, de otra forma, se tendría

$$\frac{d\mathbf{X}}{dt} - \mathbf{AX} = \mathbf{0} \tag{8.11}$$

cuya solución es del tipo

$$\mathbf{X} = \mathbf{K}e^{-\mathbf{A}t} \tag{8.12}$$

donde **K** es un vector de coeficientes que depende de las condiciones iniciales. Para este caso, si se tiene un sistema desacoplado, no hay duda de que es más sencillo que hacer la evaluación de la expresión $e^{-\mathbf{A}t}$.

www.FreeLibros.me

Se presentan a continuación algunas propiedades y resultados de la teoría de matrices, suponiendo que la matriz **A** tiene coeficientes reales:

1. Una matriz se dice que es ortogonal si $a_i^T a_j = 0$ y $a_i^T a_i = 1$ $(i \neq j)$, donde el vector a_i es la i-ésima columna de **A**. Se debe observar que la inversa de una matriz ortogonal se puede obtener sin hacer cálculos, ya que $\mathbf{A}^\mathbf{T}\mathbf{A} = \mathbf{I}$, y así $\mathbf{A}^{-1} = \mathbf{A}^\mathbf{T}$.
2. Los valores propios de una matriz simétrica real son reales. Los vectores propios correspondientes a valores propios distintos son ortogonales.
3. Si una matriz de orden n tiene n valores propios distintos, entonces hay n vectores propios linealmente independientes, los cuales pueden formar una base para el espacio vectorial. Así, un vector arbitrario se puede expresar en términos de esos vectores propios; por ejemplo, $y = \sum_{r=1}^{n} a_r \mathbf{X}^{(\mathbf{r})}$, donde $\mathbf{X}^{(\mathbf{r})}$ $(r = 1, 2, \ldots, n)$ son los vectores propios linealmente independientes.
4. Si $\mathbf{X}^{(\mathbf{i})}$ es el vector propio correspondiente al valor propio λ_i, entonces $\mathbf{AX}^{(\mathbf{i})} = \lambda_i \mathbf{X}^{(\mathbf{i})}$ y $\mathbf{A}^\mathbf{k}\mathbf{X}^{(\mathbf{i})} = \lambda_i^k \mathbf{X}^{(\mathbf{i})}$. Así, el efecto de multiplicaciones sucesivas de un vector propio por la matriz **A** es multiplicar sucesivamente el vector por el escalar λ_i.
5. Dos matrices **A** y **B** se dice que son *similares* si existe una matriz **P** no singular tal que $\mathbf{B} = \mathbf{P}^{-1}\mathbf{AP}$. Es fácil observar que dos matrices *similares* tienen los mismos valores propios debido a que, $\mathbf{AX} = \lambda\mathbf{X}$. Entonces $\mathbf{P}^{-1}\mathbf{AX} = \lambda\mathbf{P}^{-1}\mathbf{X}$ y, si $\mathbf{X} = \mathbf{PY}$, entonces $\mathbf{P}^{-1}\mathbf{APY} = \lambda\mathbf{Y}$. Así, los vectores propios de **A** se pueden encontrar a partir de los vectores propios de **B**, por la relación $\mathbf{X} = \mathbf{PY}$.

   NOTA: Las *matrices similares*, o *de similaridad*, son aquellas que tienen los mismos valores propios. Es decir, si $\mathbf{B} = \mathbf{M}^{-1}\mathbf{AM}$ entonces **A** y **B** tienen los mismos valores propios. Un vector propio **X** de **A** corresponde a un vector propio $\mathbf{M}^{-1}\mathbf{X}$ de **B**. Así, matrices *similares* representan la misma transformación respecto a diferentes bases. Resumiendo, se tiene que:

   > Las matrices **A** y **B** que representan la misma transformación lineal **T** con respecto a dos bases diferentes v y V son *similares* si
   >
   > $$[\mathbf{T}]_{V\ a\ V} = [\mathbf{I}]_{v\ a\ V}\ [\mathbf{T}]_{v\ a\ v}\ [\mathbf{I}]_{V\ a\ v} \quad \textbf{(8.13)}$$
   >
   > $$\mathbf{B} = \mathbf{M}^{-1}\ \ \mathbf{A}\ \ \mathbf{M} \quad \textbf{(8.14)}$$

6. Si $\mathbf{X}^{(\mathbf{r})}$ es un vector propio de una matriz, entonces cualquier escalar múltiplo de éste también es un vector propio. Algunas veces es conveniente normalizar el vector propio, y esto se puede hacer de dos maneras: Un método de normalización es dividir todos los elementos del vector por el elemento de módulo más grande; entonces esos vectores tienen unidad respecto al elemento más grande. Alternativamente, cada elemento se puede dividir entre la suma de los cuadrados de los elementos del vector, en cuyo caso los vectores tienen longitud unitaria.

8.2 Forma diagonal de una matriz

En una matriz cualquiera **A**, si sus vectores propios son $\mathbf{X}_i$ y se tiene de antemano que $\mathbf{AX} = \lambda\mathbf{X}$, entonces se llega a [Maron *et al.*, 1995], [Grossman, 1996]:

$$\mathbf{A}[\mathbf{X}_1 \quad \mathbf{X}_2 \quad \cdots \quad \mathbf{X}_n] = [\lambda_1\mathbf{X}_1 \quad \lambda_2\mathbf{X}_2 \quad \cdots \quad \lambda_n\mathbf{X}_n] \quad \textbf{(8.15)}$$

Reacomodando el lado derecho de la ecuación, se llega a

$$\mathbf{A}[\mathbf{X}_1 \quad \mathbf{X}_2 \quad \cdots \quad \mathbf{X}_n] = [\mathbf{X}_1 \quad \mathbf{X}_2 \quad \cdots \quad \mathbf{X}_n]\begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix} \quad \textbf{(8.16)}$$

www.FreeLibros.me

Agrupando sus vectores propios $\mathbf{X}_i$ como las columnas de **M**, se llega a la expresión $\mathbf{AM} = \mathbf{M}\Lambda$; así: resumiendo, se tiene

Suponiendo que la matriz **A** de $(n \times n)$ tiene n vectores propios linealmente independientes, si además esos vectores se eligen para ser las columnas de la matriz **M**, se tiene que $\mathbf{M}^{-1}\mathbf{AM}$ es una matriz diagonal Λ, con los valores propios de la matriz **A** a lo largo de la diagonal

$$\mathbf{M}^{-1}\mathbf{AM} = \Lambda = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix} \tag{8.17}$$

Asimismo se tiene, por tanto, que

$$\mathbf{A} = \mathbf{M}\Lambda\mathbf{M}^{-1} \tag{8.18}$$

Así se concluye que si una matriz tiene valores propios distintos, es absolutamente diagonalizable. La matriz de diagonalización **M** no es única. Esto se debe al hecho de que cualquier vector propio de **A** se puede multiplicar por una constante y sigue siendo un vector propio de **A**. De esta manera, la matriz **M** se puede multiplicar por cualquier valor diferente de cero y se obtiene una nueva matriz de diagonalización. La ecuación $\mathbf{AM} = \mathbf{M}\Lambda$ se sostiene si y sólo si las columnas de **M** son los vectores propios de **A**. Ninguna otra matriz **M** conduce a una matriz diagonal de **A**. Como no todas las matrices poseen n vectores propios linealmente independientes, se deduce que no todas las matrices son diagonalizables.

8.3 Forma canónica de Jordan

El objetivo es hacer el producto $\mathbf{M}^{-1}\mathbf{AM}$ lo más cercano a una matriz diagonal, para cualquier matriz **A** [Grossman, 1996]. Para una matriz defectiva, el resultado del esfuerzo de "diagonalizar" es la forma canónica de Jordan $\mathbf{J} = \mathbf{S}^{-1}\mathbf{AS}$, donde la matriz **J** es triangular superior. Por tanto, los valores propios quedan en la diagonal. En un caso sencillo, **A** tiene un conjunto completo de vectores propios y éstos son las columnas de **M**, lo cual lleva a la diagonalización $\mathbf{M}^{-1}\mathbf{AM} = \Lambda$. Por otro lado, para el caso general donde faltan algunos vectores propios y la forma diagonal es imposible, se llega a la forma canónica de Jordan y la matriz **M** cambia por la matriz **S**.

Si una matriz **A** tiene s vectores propios linealmente independientes, entonces se tiene una matriz en la forma canónica de Jordan con s bloques cuadrados en la diagonal, de la forma

$$\mathbf{J} = \mathbf{S}^{-1}\mathbf{AS} = \begin{bmatrix} \mathbf{J}_1 & & & \\ & \mathbf{J}_2 & & \\ & & \ddots & \\ & & & \mathbf{J}_s \end{bmatrix} \tag{8.19}$$

Si una matriz cuadrada **A** tiene valores propios repetidos, no se puede llegar a una representación en la forma diagonal. Sin embargo, se puede llegar a una forma diagonal por bloques, también llamada *forma canónica de Jordan*, la cual se obtiene de la siguiente manera:

Se considera una matriz **A** de $n \times n$ con valor propio de multiplicidad n. En otras palabras la matriz **A** sólo tiene un valor propio. Por ejemplo, si se supone $n = 4$ donde $(\mathbf{A} - \lambda\mathbf{I})$ tiene como rango $n - 1 = 3$ o en forma equivalente nulidad 1, entonces la ecuación

www.FreeLibros.me

$$(\mathbf{A}-\lambda\mathbf{I})\mathbf{X}=\mathbf{0} \tag{8.20}$$

tiene solamente una solución lineal que depende de un parámetro. Por tanto, sólo hay un vector linealmente independiente asociado a λ.

Para formar una base en R^4, se necesitan tres vectores más, linealmente independientes. Estos tres vectores $\mathbf{X}_n\,(n=2, 3, 4)$ se eligen de tal forma que cumplan con la propiedad

$$(\mathbf{A}-\lambda\mathbf{I})^n\,\mathbf{X}_n=\mathbf{0} \tag{8.21}$$

Un vector $\mathbf{V}$ se llama vector propio generalizado de grado n si

$$(\mathbf{A}-\lambda\mathbf{I})^n\,\mathbf{V}=\mathbf{0} \tag{8.22}$$

Adicionalmente, se debe cumplir que

$$(\mathbf{A}-\lambda\mathbf{I})^{n-1}\,\mathbf{V}\neq\mathbf{0} \tag{8.23}$$

Si $n=1$, el problema se reduce a $(\mathbf{A}-\lambda\mathbf{I})\mathbf{V}=\mathbf{0}$ con $\mathbf{V}\neq\mathbf{0}$, donde $\mathbf{V}$ es un vector propio ordinario. Para el caso anterior donde $n=4$, se tiene que

$$\mathbf{V}_4:=\mathbf{V} \tag{8.24a}$$

$$\mathbf{V}_3:=(\mathbf{A}-\lambda\mathbf{I})\mathbf{V}_4=(\mathbf{A}-\lambda\mathbf{I})\mathbf{V} \tag{8.24b}$$

$$\mathbf{V}_2:=(\mathbf{A}-\lambda\mathbf{I})\mathbf{V}_3=(\mathbf{A}-\lambda\mathbf{I})^2\,\mathbf{V} \tag{8.24c}$$

$$\mathbf{V}_1:=(\mathbf{A}-\lambda\mathbf{I})\mathbf{V}_2=(\mathbf{A}-\lambda\mathbf{I})^3\,\mathbf{V} \tag{8.24d}$$

A esto se le llama cadena de vectores propios generalizados de longitud $n=4$, los cuales tienen la propiedad de que

$$(\mathbf{A}-\lambda\mathbf{I})\mathbf{V}=\mathbf{0} \tag{8.25a}$$

$$(\mathbf{A}-\lambda\mathbf{I})^2\,\mathbf{V}_2=\mathbf{0} \tag{8.25b}$$

$$(\mathbf{A}-\lambda\mathbf{I})^3\,\mathbf{V}_3=\mathbf{0} \tag{8.25c}$$

$$(\mathbf{A}-\lambda\mathbf{I})^4\,\mathbf{V}_4=\mathbf{0} \tag{8.25d}$$

Estos vectores, como se generan en forma automática, son linealmente independientes y, por tanto, se pueden usar como base. De las ecuaciones anteriores se puede deducir

$$\mathbf{AV}_1=\lambda\mathbf{V}_1 \tag{8.26a}$$

$$\mathbf{AV}_2=\mathbf{V}_1+\lambda\mathbf{V}_2 \tag{8.26b}$$

$$\mathbf{AV}_3=\mathbf{V}_2+\lambda\mathbf{V}_3 \tag{8.26c}$$

$$\mathbf{AV}_4=\mathbf{V}_3+\lambda\mathbf{V}_4 \tag{8.26d}$$

EJEMPLO 8.1

Ejercicio numérico para el caso de valores propios repetidos. Si se tiene la siguiente matriz

$$\mathbf{A}=\begin{bmatrix} \mathbf{8} & 0 & 0 & \mathbf{8} & \mathbf{8} \\ 0 & 0 & 0 & \mathbf{8} & \mathbf{8} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{8} \end{bmatrix}$$

www.FreeLibros.me

Los valores propios de esta matriz son $\lambda_1 = 8,\ \lambda_2 = 0,\ \lambda_3 = 0,\ \lambda_4 = 0$ y $\lambda_5 = 8$.

Del primer valor propio se calcula el primer vector propio con la relación $\mathbf{AV}_1 = \lambda \mathbf{V}_1$. Así se forma el siguiente sistema de ecuaciones

$$\begin{bmatrix} \mathbf{8} & 0 & 0 & \mathbf{8} & \mathbf{8} \\ 0 & 0 & 0 & \mathbf{8} & \mathbf{8} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{8} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \end{bmatrix} = 8 \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \end{bmatrix}$$

En forma expandida, se tiene por tanto que

$$8X_1 + 8X_4 + 8X_5 = 8X_1$$

$$8X_4 + 8X_5 = 8X_2$$

$$0 = 8X_3$$

$$0 = 8X_4$$

$$8X_5 = 8X_5$$

La solución a este sistema de ecuaciones es la siguiente:

1. De la ecuación 3 se determina que $X_3 = 0$
2. De la ecuación 4 se determina que $X_4 = 0$
3. Sustituyendo $X_4 = 0$ en la ecuación (1), se determina que $X_5 = 0$
4. Sustituyendo $X_4 = 0$ y $X_5 = 0$ en la ecuación (2), se obtiene que $X_2 = 0$
5. Por tanto, la única solución no trivial es que X_1 tome cualquier valor, por ejemplo $X_1 = 1$

Por tanto, el primer vector propio es

$$\mathbf{V}_1' = [1 \quad 0 \quad 0 \quad 0 \quad 0]$$

El último valor propio repetido es $\lambda_5 = 8$, por lo que su vector propio se calcula con la ecuación de la forma (26b), o sea $\mathbf{AV}_5 = \mathbf{V}_1 + \lambda \mathbf{V}_5$. Con esto se forma el sistema de ecuaciones siguiente:

$$8X_1 + 8X_4 + 8X_5 = 1 + 8X_1$$

$$8X_4 + 8X_5 = 0 + 8X_2$$

$$0 = 0 + 8X_3$$

$$0 = 0 + 8X_4$$

$$8X_5 = 0 + 8X_5$$

La solución a este sistema de ecuaciones es:

1. De la ecuación 3 se determina que $X_3 = 0$
2. De la ecuación 4 se determina que $X_4 = 0$
3. Despejando de la ecuación 1 a X_5 y sustituyendo el valor de $X_4 = 0$, se encuentra que $X_5 = \frac{1}{8}$
4. De la ecuación 2, se obtiene que $X_2 = X_5$. Por tanto, $X_2 = \frac{1}{8}$

www.FreeLibros.me

5. Finalmente, de la ecuación 1 se determina que X_1 puede tomar cualquier valor, siempre y cuando se cumpla con la restricción que determina la ecuación (25b), es decir $(\mathbf{A}-\lambda\mathbf{I})^2\mathbf{V}_5=\mathbf{0}$, lo cual se cumple con $X_1=1$

Por tanto, el quinto vector propio es

$$\mathbf{V}_5'=\begin{bmatrix}1 & \frac{1}{8} & 0 & 0 & \frac{1}{8}\end{bmatrix}$$

Del segundo valor propio se calcula el segundo vector propio con la relación $\mathbf{AV}_2=\lambda\mathbf{V}_2$. Así se forma el siguiente sistema de ecuaciones:

$$\begin{aligned}8X_1+8X_4+8X_5&=0X_1\\8X_4+8X_5&=0X_2\\0&=0X_3\\0&=0X_4\\8X_5&=0X_5\end{aligned}$$

La solución a este sistema de ecuaciones es:

1. De la ecuación 5 se determina que $X_5=0$
2. Al sustituir este valor en la ecuación 2, se determina que $X_4=0$
3. Sustituyendo estos valores en la ecuación 1, se determina que $X_1=0$
4. Como no hay restricción para X_2 y X_3, se pueden obtener tres soluciones, pero sólo dos de ellas son linealmente independientes. Así se tiene que una de ellas es cuando $X_2=1$ y $X_3=0$, y la otra cuando $X_2=0$ y $X_3=1$. Por tanto, se tiene que la que sirva para calcular $\mathbf{V}_3$ será la que se toma como $\mathbf{V}_2$, y la otra como $\mathbf{V}_4$. Así se tiene que

$$\mathbf{V}_2'=\begin{bmatrix}0 & 1 & 0 & 0 & 0\end{bmatrix} \text{ y } \mathbf{V}_4'=\begin{bmatrix}0 & 0 & 1 & 0 & 0\end{bmatrix}$$

El tercer valor propio se repite, por lo que su vector propio se calcula con la ecuación de la forma (8.26b). Si se utiliza $\mathbf{V}_4$ como vector propio auxiliar, es decir $\mathbf{AV}_3=\mathbf{V}_4+\lambda\mathbf{V}_3$, se tiene que la tercera ecuación es $0=1+0X_3$, imposible de cumplir. Por esta razón se utiliza $\mathbf{V}_2$ como vector propio auxiliar, es decir $\mathbf{AV}_3=\mathbf{V}_2+\lambda\mathbf{V}_3$.

Así se forma el sistema de ecuaciones siguiente:

$$\begin{aligned}8X_1+8X_4+8X_5&=0+0X_1\\8X_4+8X_5&=1+0X_2\\0&=0+0X_3\\0&=0+0X_4\\8X_5&=0+0X_5\end{aligned}$$

La solución a este sistema de ecuaciones es:

1. Por la ecuación 5, se determina que $X_5=0$
2. Por la ecuación 2, se determina que $X_4=\frac{1}{8}$

www.FreeLibros.me

3. De la ecuación 1 se deduce que $X_1 = -X_4$. Por tanto, se tiene que $X_1 = -\frac{1}{8}$
4. Como no hay restricción para X_2 y X_3, siempre y cuando se cumpla que $(\mathbf{A} - \lambda\mathbf{I})^2 \mathbf{V}_3 = \mathbf{0}$, los valores pueden ser $X_2 = 1$ y $X_3 = 1$

Por tanto, el tercer vector propio es

$$\mathbf{V}_3' = \begin{bmatrix} -\frac{1}{8} & 1 & 1 & \frac{1}{8} & 0 \end{bmatrix}$$

Agrupando todos los vectores propios, se forma la matriz de diagonalización y su inversa como sigue:

$$\mathbf{S} = \begin{bmatrix} 1 & 0 & -\frac{1}{8} & 0 & 1 \\ 0 & 1 & 1 & 0 & \frac{1}{8} \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & \frac{1}{8} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{8} \end{bmatrix} \quad \mathbf{S}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 1 & \mathbf{-8} \\ 0 & 1 & 0 & \mathbf{-8} & -1 \\ 0 & 0 & 0 & \mathbf{8} & 0 \\ 0 & 0 & 1 & \mathbf{-8} & 0 \\ 0 & 0 & 0 & 0 & \mathbf{8} \end{bmatrix}$$

De aquí se obtiene la forma canónica de Jordan

$$\mathbf{J} = \mathbf{S}^{-1}\mathbf{AS} = \begin{bmatrix} 1 & 0 & 0 & 1 & \mathbf{-8} \\ 0 & 1 & 0 & \mathbf{-8} & -1 \\ 0 & 0 & 0 & \mathbf{8} & 0 \\ 0 & 0 & 1 & \mathbf{-8} & 0 \\ 0 & 0 & 0 & 0 & \mathbf{8} \end{bmatrix} \begin{bmatrix} \mathbf{8} & 0 & 0 & \mathbf{8} & \mathbf{8} \\ 0 & 0 & 0 & \mathbf{8} & \mathbf{8} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{8} \end{bmatrix} \begin{bmatrix} 1 & 0 & -\frac{1}{8} & 0 & 1 \\ 0 & 1 & 1 & 0 & \frac{1}{8} \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & \frac{1}{8} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{8} \end{bmatrix}$$

de donde se obtiene la matriz **J**

$$\mathbf{J} = \mathbf{S}^{-1}\mathbf{AS} = \begin{bmatrix} \mathbf{8} & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{8} \end{bmatrix}$$

Si se quiere darle acomodo a la forma canónica de Jordan, se tiene que cambiar el renglón 5 al dos y la columna 5 se pone en columna 2 y se recorren las demás de tal forma que

$$\mathbf{J} = \mathbf{S}^{-1}\mathbf{AS} = \begin{bmatrix} \mathbf{J}_1 & & & \\ & \mathbf{J}_2 & & \\ & & \ddots & \\ & & & \mathbf{J}_s \end{bmatrix} = \begin{bmatrix} \mathbf{8} & 1 & & & \\ 0 & \mathbf{8} & & & \\ & & 0 & 1 & \\ & & 0 & 0 & \\ & & & & 0 \end{bmatrix}$$

www.FreeLibros.me

8.4 Potencias de una matriz

Suponiendo que se tienen los valores propios y los vectores propios de la matriz **A**, entonces los valores propios de $\mathbf{A}^2$ son exactamente $\lambda_1^2, \lambda_2^2, \ldots, \lambda_n^2$, y cada vector propio de **A** también es un vector propio de $\mathbf{A}^2$. Así, si se parte de la expresión $\mathbf{A}=\mathbf{M}\Lambda\mathbf{M}^{-1}$, se tiene, por tanto [Grossman, 1996]

$$\mathbf{A}^2=\left(\mathbf{M}\Lambda\mathbf{M}^{-1}\right)\left(\mathbf{M}\Lambda\mathbf{M}^{-1}\right) \tag{8.27}$$

En esta expresión, el producto $\mathbf{M}^{-1}$ por **M** se cancela, para dar finalmente

$$\mathbf{A}^2=\mathbf{M}\Lambda^2\mathbf{M}^{-1} \tag{8.28}$$

En resumen se tendrá:

> Los valores propios de $\mathbf{A}^k$ son $\lambda_1^k, \lambda_2^k, \ldots, \lambda_n^k$ a la k-ésima potencia de los valores propios de la matriz **A**. Cada vector propio de **A** también es un vector propio de $\mathbf{A}^k$, y si la matriz **M** "diagonaliza" la matriz **A** y también la matriz $\mathbf{A}^k$, así
>
> $$\Lambda^k=\left(\mathbf{M}^{-1}\mathbf{A}\mathbf{M}\right)\left(\mathbf{M}^{-1}\mathbf{A}\mathbf{M}\right)\cdots\left(\mathbf{M}^{-1}\mathbf{A}\mathbf{M}\right)=\mathbf{M}^{-1}\mathbf{A}^k\mathbf{M} \tag{8.29}$$
>
> Cada $\mathbf{M}^{-1}$ cancela una **M**, excepto por la primera $\mathbf{M}^{-1}$ y la última **M**.
>
> Si la matriz **A** es invertible, esta regla también se aplica a su inversa, donde $k=-1$, así, los valores propios de $\mathbf{A}^{-1}$ son $\frac{1}{\lambda_i}$.

8.5 Ecuaciones diferenciales

Cuando se tiene un sistema de ecuaciones diferenciales de la forma (8.11), es decir

$$\frac{d\mathbf{X}}{dt}-\mathbf{A}\mathbf{X}=\mathbf{0} \tag{8.30}$$

cuya solución depende de la evaluación de la exponencial de **A**, hay varias posibilidades de resolver este problema; todas conducen al mismo resultado. Las dos posibilidades más útiles se basan en: dar la definición directa de la exponencial de una matriz; la otra posibilidad es darle la interpretación física a la ecuación diferencial y a su solución mediante el uso de la diagonalización [Strang, 1988], [Grossman, 1996]. Éste es el tipo de sistemas de ecuaciones diferenciales que tiene amplias aplicaciones.

Para una matriz general **A**, lo natural es imitar o seguir la definición de series de potencia

$$e^x=1+x+\frac{x^2}{2!}+\frac{x^3}{3!}+\cdots \tag{8.31}$$

Sustituyendo la variable x por $\mathbf{A}t$, y el uno por la matriz identidad, entonces se llega a

$$e^{-\mathbf{A}t}=\mathbf{I}+\mathbf{A}t+\frac{(\mathbf{A}t)^2}{2!}+\frac{(\mathbf{A}t)^3}{3!}+\cdots \tag{8.32}$$

La ecuación diferencial tiene la solución $\mathbf{X}=\mathbf{K}e^{-\mathbf{A}t}$. Así, diagonalizando la matriz **A** y sustituyéndola en la ecuación (8.32), se llega a

$$e^{-\mathbf{A}t}=\mathbf{I}+\left(\mathbf{M}\Lambda\mathbf{M}^{-1}\right)t+\frac{\left(\mathbf{M}\Lambda^2\mathbf{M}^{-1}\right)t^2}{2!}+\frac{\left(\mathbf{M}\Lambda^3\mathbf{M}^{-1}\right)t^3}{3!}+\cdots \tag{8.33}$$

Esto conduce a

$$e^{-\mathbf{A}t}=\mathbf{M}\left(\mathbf{I}+\Lambda t+\frac{(\Lambda t)^2}{2!}+\frac{(\Lambda t)^3}{3!}+\cdots\right)\mathbf{M}^{-1} \tag{8.34}$$

www.FreeLibros.me

En la ecuación anterior, la expresión entre paréntesis corresponde a la evaluación de la exponencial de $e^{-\Lambda t}$, por lo que sustituyéndola se llega finalmente a

$$e^{-\mathbf{A}t} = \mathbf{M}e^{-\Lambda t}\mathbf{M}^{-1} \quad \textbf{(8.35)}$$

Resumiendo, un sistema de ecuaciones diferenciales tiene la siguiente solución:

Si $\mathbf{A}$ se puede diagonalizar de la forma $\mathbf{A} = \mathbf{M}\Lambda\mathbf{M}^{-1}$, entonces la ecuación diferencial $\frac{d\mathbf{u}}{dt} = \mathbf{A}\mathbf{u}$ tiene la solución

$$\mathbf{u}(t) = \mathbf{u}(0)e^{-\mathbf{A}t} = \mathbf{u}(0)\mathbf{M}e^{-\Lambda t}\mathbf{M}^{-1} \quad \textbf{(8.36)}$$

donde $\mathbf{u}(0)$ depende de las condiciones iniciales.

La estabilidad de una ecuación diferencial sólo depende de la parte real de los valores propios. De esta forma se tiene que la ecuación diferencial $\frac{d\mathbf{u}}{dt} = \mathbf{A}\mathbf{u}$ es:

Estable y $e^{\mathbf{A}t} \to 0$ siempre que todas las partes reales de los valores propios sean menores que cero, $\text{Re}(\lambda_i) < 0$.

- Marginalmente estable cuando todas las partes reales de los valores propios sean menores o iguales que cero, $\text{Re}(\lambda_i) \leq 0$, donde en algunos casos se tiene que $\text{Re}(\lambda_i) = 0$.
- Inestable y $e^{\mathbf{A}t}$ crece sin fin si algún valor propio tiene parte real positiva, $\text{Re}(\lambda_i) > 0$.

8.6 Teorema de Cayley-Hamilton

Puesto que una matriz cuadrada no nula, es semejante a una matriz diagonal, la ecuación $p(\mathbf{X}) = \mathbf{0}$ siempre es resoluble [Grossman, 1996]. De aquí se infiere el siguiente teorema:

Teorema 8.2 Cualquier matriz cuadrada $\mathbf{A}$ de orden n satisface una ecuación polinomial cuyo orden es, como máximo, n. En efecto, para cualquier matriz cuadrada $\mathbf{A}$, siempre hay una ecuación polinomial de orden n a la que satisface $\mathbf{A}$.

Demostración Para demostrar el teorema anterior, es necesario definir lo que se entiende por valor de un polinomio cuyos coeficientes no son escalares, sino matrices cuadradas, por ejemplo

$$F(\lambda) = C_0 + C_1\lambda + \cdots + C_k\lambda^k \quad \textbf{(8.37)}$$

Cuando se sustituye la variable escalar λ por la matriz cuadrada $\mathbf{A}$, como la multiplicación matricial no es conmutativa, es claro que, en general, las distintas potencias de $\mathbf{A}$ no se conmutarán con las matrices coeficiente de $F(\lambda)$.

Teorema 8.3 Si $F(\lambda)$ y $P(\lambda)$ son polinomios en la variable escalar λ con coeficientes que son matrices cuadradas, y si $P(\lambda) = F(\lambda)(\mathbf{A} - \lambda\mathbf{I})$, entonces $p(\mathbf{A}) = 0$.

Demostración Suponiendo que

$$F(\lambda) = \mathbf{C}_0 + \mathbf{C}_1\lambda + \cdots + \mathbf{C}_k\lambda^k, \quad \textbf{(8.38)}$$

www.FreeLibros.me

donde $\mathbf{C}_0, \mathbf{C}_1 \ldots \mathbf{C}_k$ son matrices de $n \times n$, entonces

$$P(\lambda) = \left(\mathbf{C}_0 + \mathbf{C}_1\lambda + \cdots + \mathbf{C}_k\lambda^k\right)(\mathbf{A} - \lambda\mathbf{I}) \tag{8.39}$$

Desarrollando la ecuación, se llega a

$$P(\lambda) = \mathbf{C}_0\mathbf{A} + \mathbf{C}_1\mathbf{A}\lambda + \cdots + \mathbf{C}_k\mathbf{A}\lambda^k - \mathbf{C}_0\lambda - \mathbf{C}_1\lambda^2 - \cdots - \mathbf{C}_k\lambda^{k+1} \tag{8.40}$$

Sustituyendo λ por $\mathbf{A}$, se obtiene

$$P(\lambda) = \mathbf{C}_0\mathbf{A} + \mathbf{C}_1\mathbf{A}^2 + \cdots + \mathbf{C}_k\mathbf{A}^{k+1} - \mathbf{C}_0\mathbf{A} - \mathbf{C}_1\mathbf{A}^2 - \cdots - \mathbf{C}_k\mathbf{A}^{k+1} = 0 \tag{8.41}$$

Teorema de Cayley-Hamilton Toda matriz cuadrada satisface su propia ecuación característica.

Demostración Sea $\mathbf{A}$ una matriz cuadrada de $n \times n$ cuya ecuación característica es

$$|\mathbf{A} - \lambda\mathbf{I}| = \begin{vmatrix} a_{11} - \lambda & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} - \lambda & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{11} & \cdots & a_{nn} - \lambda \end{vmatrix} \tag{8.42}$$

Se tiene, por la definición de matrices, que

$$[adj(\mathbf{A} - \lambda\mathbf{I})](\mathbf{A} - \lambda\mathbf{I}) = |\mathbf{A} - \lambda\mathbf{I}|\mathbf{I} = (-1)^n\left[\lambda^n\mathbf{I} - \beta_1\lambda^{n-1}\mathbf{I} + \cdots + (-1)^n\beta_n\mathbf{I}\right] \tag{8.43}$$

o bien

$$(-1)^n\left[\lambda^n\mathbf{I} - \beta_1\lambda^{n-1}\mathbf{I} + \cdots + (-1)^n\beta_n\mathbf{I}\right] = F(\lambda)(\mathbf{A} - \lambda\mathbf{I}) \tag{8.44}$$

Ésta es una relación en λ con coeficientes matriciales. Así, el primer miembro debe anularse cuando se reemplaza λ por la matriz $\mathbf{A}$. Por tanto, se llega a

$$\mathbf{A}^n - \beta_1\mathbf{A}^{n-1} + \cdots + (-1)^n\beta_n\mathbf{I} = \mathbf{0} \tag{8.45}$$

Es decir, la matriz $\mathbf{A}$ satisface su propia ecuación característica.

8.7 Cálculo de valores propios y vectores propios

Para desacoplar un sistema de ecuaciones diferenciales ordinarias, el problema se reduce al cálculo de vectores propios y valores propios de la matriz de acoplamiento. Existen muchos métodos para encontrar estos vectores propios; entre ellos, los más comunes son:

a) Método de Jacobi
b) Método de Given
c) Método de Householder
d) Multiplicación sucesiva por $\mathbf{y}^{(\mathbf{k})}$

www.FreeLibros.me

e) Método de potenciación
f) Métodos **L**-**R** y **Q**-**R**

Todos ellos, con diferentes principios y metodologías, tratan de encontrar todo el conjunto de valores propios y vectores propios. A continuación se da una descripción de cada uno de ellos.

8.7.1 Método de Jacobi

Los métodos de Jacobi usan transformaciones de similitud para obtener una matriz transformada con los mismos valores propios, pero con una estructura simple. Las matrices de transformación que se usan son matrices ortogonales. Se puede demostrar que éstas son las más adecuadas para la minimización del error en el proceso. La forma más sencilla que se puede lograr es la forma diagonal, ya que los valores propios estarán disponibles directamente en los elementos diagonales. El método de Jacobi está ideado para dar una forma diagonal eliminando sistemáticamente los elementos fuera de la diagonal [Mathews *et al.*, 2000], [Burden *et al.*, 2002], [Nieves *et al.*, 2002]. Esto, sin embargo, es un proceso iterativo que requiere un gran número de pasos. Esto presenta dos desventajas: primero, el proceso puede converger lentamente o no converger; segundo, la necesidad de truncar el proceso puede producir errores que alteren seriamente la solución correcta.

El esquema computacional es muy sencillo. Se forma una nueva matriz $\mathbf{A}_1 = \mathbf{P}_1^{-1}\mathbf{A}\mathbf{P}_1$ utilizando una matriz $\mathbf{P}_1$ que introduce un elemento cero fuera de la diagonal en $\mathbf{A}_1$. Se produce una matriz más $\mathbf{A}_2$ como $\mathbf{A}_2 = \mathbf{P}_2^{-1}\mathbf{A}_1\mathbf{P}_2$, en la cual se produce un nuevo cero fuera de la diagonal. Desafortunadamente, en el proceso de Jacobi, la introducción de cada nuevo cero, en general, introduce un nuevo elemento dentro de las posiciones previas con ceros. El proceso continúa trabajando sistemáticamente a lo largo de un renglón y luego el siguiente, introduciendo elementos ceros, o de manera alterna, eliminando el elemento fuera de la diagonal de módulo más grande en cada etapa. El proceso termina cuando todos los elementos fuera de la diagonal son menores en módulo que alguna cantidad pequeña especificada. De esta manera, los valores propios son los elementos de la diagonal.

Debido a que el cálculo computacional se lleva a cabo con matrices ortogonales, no se necesita calcular la inversa, que está dada por $\mathbf{P}_1^{-1} = \mathbf{P}_1^{T}$. La forma final de la matriz es

$$\mathbf{A}_\mathbf{r} = \mathbf{P}_\mathbf{r}^\mathbf{T}\mathbf{P}_{\mathbf{r}-1}^\mathbf{T}\cdots\mathbf{P}_1^\mathbf{T}\mathbf{A}\mathbf{P}_1\cdots\mathbf{P}_{\mathbf{r}-1}\mathbf{P}_\mathbf{r} \tag{8.46}$$

y, si $\mathbf{Y}^{(\mathbf{r})}$ es el vector propio de $\mathbf{A}_\mathbf{r}$, el vector propio de la matriz original es

$$\mathbf{P}_1\cdots\mathbf{P}_{\mathbf{r}-1}\mathbf{P}_\mathbf{r}\mathbf{Y}^{(\mathbf{r})} \tag{8.47}$$

Las matrices ortogonales que se usan en el método de Jacobi son extensiones de una matriz rotacional en dos dimensiones. La matriz rotacional $(n\times n)$ para rotar en el plano (r, s) está determinada por matrices unitarias con los siguientes cuatro cambios.

$$a_{rr} = \cos\theta \tag{8.48a}$$

$$a_{rs} = -\operatorname{sen}\theta \tag{8.48b}$$

$$a_{sr} = \operatorname{sen}\theta \tag{8.48c}$$

$$a_{ss} = \cos\theta \tag{8.48d}$$

Por ejemplo, la siguiente matriz corresponde a una rotación en el plano $(2, 3)$.

$$\mathbf{P}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c & -s & 0 \\ 0 & s & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \begin{aligned} c &= \cos\theta \\ s &= \operatorname{sen}\theta \end{aligned} \tag{8.49}$$

www.FreeLibros.me

La transformación $\mathbf{P}_1^{\mathrm{T}}\mathbf{A}\mathbf{P}_1$ da la matriz

$$\begin{bmatrix} a_{11} & ca_{12}+sa_{13} & -sa_{12}+ca_{13} & a_{14} \\ ca_{21}+sa_{31} & c^2a_{22}+csa_{23}+csa_{32}+s^2a_{33} & -csa_{22}+c^2a_{23}-s^2a_{32}+csa_{33} & ca_{24}+sa_{34} \\ -sa_{21}+ca_{31} & -csa_{22}-s^2a_{23}+c^2a_{32}+csa_{33} & s^2a_{22}-csa_{23}-sca_{33}+c^2a_{32} & ca_{34}-sa_{24} \\ a_{41} & ca_{42}+sa_{43} & -sa_{42}+ca_{43} & a_{44} \end{bmatrix} \quad \textbf{(8.50)}$$

El método de Jacobi reduce un coeficiente a cero escogiendo un valor de θ tal que el elemento de la posición $(2, 3)$ se haga cero. El método se usa normalmente para matrices simétricas, por lo que se requiere

$$\left(c^2-s^2\right)a_{23}+cs\left(a_{33}-a_{22}\right)=0 \quad \textbf{(8.51)}$$

o bien

$$\tan 2\theta=\frac{2a_{23}}{a_{22}-a_{33}} \quad \textbf{(8.52)}$$

8.7.2 Método de Given

El método de Given se basa en una transformación matricial del mismo tipo que el método de Jacobi; pero el esquema está diseñado de tal forma que ningún cero que se cree se retenga (cambie) en las transformaciones subsecuentes. Cuando se realiza la rotación en el plano (r, s) se elimina el elemento $(r-1, s)$ para $(r=1, 2,\ldots, n-1)$ y $(s=r+2, r+3,\ldots, n)$. Entonces, para la rotación anterior en el plano $(2, 3)$, se elige el valor de θ para satisfacer

$$-sa_{12}+ca_{13}=0 \quad \textbf{(8.53)}$$

$$\tan\theta=\frac{a_{13}}{a_{12}} \quad \textbf{(8.54)}$$

o, en forma más general,

$$\tan\theta=\frac{a_{r-1,s}}{a_{r-1,r}} \quad \textbf{(8.55)}$$

Se observa que los elementos de la diagonal principal, y los elementos inmediatamente arriba y abajo, permanecen diferentes de cero. Entonces, el resultado final del proceso no es una forma *diagonal simple*, sino la llamada *forma tridiagonal*.

$$\begin{bmatrix} x & x & & & & & 0 & 0 & 0 \\ x & x & x & & & & & 0 & 0 \\ & x & x & x & & & & & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & & & & & x & x & x & \\ 0 & 0 & & & & & x & x & x \\ 0 & 0 & 0 & & & & & x & x \end{bmatrix} \quad \textbf{(8.56)}$$

Los valores propios de una forma tridiagonal no se pueden encontrar de manera automática, como en el caso de la forma diagonal. Sin embargo, el método de solución es bastante sencillo. Esto hace que en la reducción a la forma tridiagonal valga la pena el empleo de una computadora. Es fácil ver que los ceros se conservan si la eliminación se hace de manera sistemática en todo el primer renglón, comenzando por el elemento $(1, 3)$, y a continuación en todo el segundo renglón, empezando con el elemento $(2, 4)$, etcétera.

www.FreeLibros.me

Si la matriz **A** es no simétrica, el método de Given se puede seguir aplicando. Sin embargo, la forma final no será la forma tridiagonal simétrica, sino la forma Hessenberg,

$$\begin{bmatrix} x & x & & & & & 0 & 0 & 0 \\ x & x & x & & & & & 0 & 0 \\ x & x & x & x & & & & & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ x & x & x & x & x & x & x & x & \\ x & x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x & x \end{bmatrix} \tag{8.57}$$

Aunque el método anterior efectúa una simplificación considerable en la forma de la matriz, esto tendrá poco valor, a menos que la matriz tridiagonal resultante tenga una forma apropiada para una solución fácil. De hecho, la forma tridiagonal conduce a la secuencia Sturm, la cual es fácil de calcular. Entonces, puede ser fácil encontrar una aproximación a las raíces y después precisarlas, por ejemplo, con el método de Newton. La secuencia Sturm se genera con una secuencia recursiva como se describe a continuación. Si se deja que $f_r(\lambda)$ sea el valor del determinante

$$\begin{Vmatrix} a_1-\lambda & b_2 & & & 0 \\ b_2 & a_2-\lambda & b_3 & & \\ & b_3 & a_3-\lambda & & \\ & & & \ddots & b_r \\ 0 & & & b_r & a_r-\lambda \end{Vmatrix} \tag{8.58}$$

el cual define la ecuación característica de una matriz tridiagonal. Extendiendo el determinante por la última columna, resulta

$$f_r(\lambda)=(a_r-\lambda)f_{r-1}(\lambda)-b_r^2 f_{r-2}(\lambda) \tag{8.59}$$

lo cual es cierto para $r=n, n-1,\ldots, 2$.

Si se define

$$f_0(\lambda)=1 \tag{8.60a}$$

$$f_1(\lambda)=(a_1-\lambda), \tag{8.60b}$$

entonces las ecuaciones (8.59) para $r=2, 3,\ldots, n$ y (8.60) definen una secuencia de valores que es la secuencia Sturm. El número de cambios en signo de esta secuencia se puede tabular para varios valores de λ y determinar la posición aproximada de las raíces.

Así, es posible utilizar un método iterativo adecuado para encontrar con bastante precisión (ejemplo, $\varepsilon_r=(10)^{-3}$) el valor de una raíz. En esta forma las raíces individuales, o todas las raíces, se pueden encontrar de acuerdo con los requisitos del problema. El proceso es computacionalmente conveniente, ya que el proceso para calcular la secuencia Sturm también produce, como un término de la secuencia, el valor de la función $f_n(\lambda)$ necesario en la iteración de la ecuación.

8.7.3 Método de Householder

Aunque el método de Given tiene un avance considerable respecto al método de Jacobi, éste se ha reemplazado por el método de Householder [Nakamura, 1992], [Maron *et al.*, 1995], [Burden *et al.*, 2002]. Este

www.FreeLibros.me

método también usa transformaciones ortogonales para reducir una matriz simétrica a una matriz similar con la forma tridiagonal simétrica. La ventaja del método es que todos los posibles ceros en un reglón se producen con una sola transformación. El método Householder, por tanto, emplea $(n-2)$ transformaciones de similitud comparado con $\left[\frac{n^2-3n+2}{2}\right]$ para el método de Given. El método Householder es computacionalmente más complicado, pero queda todavía una reducción sustancial en tiempo de cómputo. Asimismo, la disminución del número de operaciones de cómputo reduce el error propagado.

Teorema (de reflexión de Householder) 8.4

Si X y Y son vectores que tienen la misma norma, entonces existe una matriz ortogonal simétrica tal que

$$\mathbf{Y} = \mathbf{PX} \tag{8.61}$$

donde

$$\mathbf{P}^{(k)} = \mathbf{I} - 2\mathbf{W}^{(k)}\left[\mathbf{W}^{(k)}\right]^{\mathrm{T}}, \qquad \left[\mathbf{W}^{(k)}\right]^{\mathrm{T}}\mathbf{W}^{(k)} = 1$$

con

$$\mathbf{W}^{(k)} = \frac{\mathbf{X}-\mathbf{Y}}{\|\mathbf{X}-\mathbf{Y}\|} \tag{8.62}$$

debido a que $\mathbf{P_r}$ es ortogonal y simétrica. Por tanto, se tiene que $\mathbf{P}^{-1} = \mathbf{P}$.

Corolario (*k*-ésima matriz de Householder)

Si se deja que **A** sea una matriz de $(n \times n)$ y **X** cualquier vector, si k es un número entero entre $1 \le k \le n-2$, se puede construir un vector $\mathbf{W}^{(k)}$ y una matriz $\mathbf{P}^{(k)} = \mathbf{I} - 2\mathbf{W}^{(k)}\left[\mathbf{W}^{(k)}\right]^{\mathrm{T}}$ tales que

$$\mathbf{P}^{(k)}\mathbf{X} = \mathbf{P}^{(k)}\begin{bmatrix} x_1 \\ \vdots \\ x_k \\ x_{k+1} \\ x_{k+2} \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_1 \\ \vdots \\ x_k \\ -S \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \mathbf{Y} \tag{8.63}$$

Suponiendo que **A** es una matriz simétrica de $(n \times n)$, entonces una secuencia de $n-2$ transformaciones de la forma **PAP** reduce la matriz **A** a una matriz simétrica tridiagonal, la cual es similar a la matriz **A**. La primera transformación está definida por $\mathbf{A}_1 = \mathbf{P}_1\mathbf{A}\mathbf{P}_1$, donde la matriz $\mathbf{P}_1$ se construye a partir del corolario, con el vector **X** igual a la primera columna de la matriz **A**. En forma general, $\mathbf{P}_1$ tiene la siguiente estructura:

$$\mathbf{P}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & p_{22} & p_{23} & p_{24} & \cdots & p_{2n} \\ 0 & p_{32} & p_{33} & & & \\ 0 & p_{42} & & \ddots & & \\ \vdots & \vdots & & & \ddots & \\ 0 & p_{n2} & & & & p_{nn} \end{bmatrix}$$

www.FreeLibros.me

Como resultado, la transformación $\mathbf{A}_1 = \mathbf{P}_1\mathbf{A}\mathbf{P}_1$ no afecta el elemento a_{11} de la matriz $\mathbf{A}$. La matriz resultante tiene la forma

$$\mathbf{A}_1 = \mathbf{P}_1\mathbf{A}\mathbf{P}_1 = \begin{bmatrix} a_{11} & u_1 & 0 & 0 & \cdots & 0 \\ u_1 & z_1 & w_{23} & w_{24} & \cdots & w_{2n} \\ 0 & w_{32} & w_{33} & & & \\ 0 & w_{42} & & \ddots & & \\ \vdots & \vdots & & & \ddots & \\ 0 & w_{n2} & & & & w_{nn} \end{bmatrix}$$

debido a que el vector $\mathbf{X}$ es la primera columna de $\mathbf{A}$, de acuerdo con la ecuación (8.63). Esto implica que $u_1 = -S$. La segunda transformación de Householder se aplica a la matriz $\mathbf{A}_1$, donde $\mathbf{P}_2$ se construye aplicando el corolario con el vector $\mathbf{X}$, igual a la segunda columna de la matriz $\mathbf{A}_1$. En forma general, $\mathbf{P}_2$ es

$$\mathbf{P}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & p_{33} & p_{34} & & p_{3n} \\ 0 & 0 & p_{43} & p_{44} & & \\ \vdots & \vdots & & & \ddots & \\ 0 & 0 & p_{n3} & & & p_{nn} \end{bmatrix}$$

El bloque de identidad de (2×2) en la parte superior izquierda garantiza que la tridiagonalización lograda en el paso anterior no se altere por la siguiente transformación, $\mathbf{A}_2 = \mathbf{P}_2\mathbf{A}_1\mathbf{P}_2$.

La matriz resultante tiene la forma siguiente:

$$\mathbf{A}_2 = \mathbf{P}_2\mathbf{A}_1\mathbf{P}_2 = \begin{bmatrix} a_{11} & u_1 & 0 & 0 & \cdots & 0 \\ u_1 & z_1 & u_2 & 0 & \cdots & 0 \\ 0 & u_2 & z_2 & w_{34} & & w_{3n} \\ 0 & 0 & w_{43} & w_{44} & & \\ \vdots & \vdots & & & \ddots & \\ 0 & 0 & w_{n3} & & & w_{nn} \end{bmatrix}$$

La $(n-2)$ transformación de Householder se aplica a la matriz $\mathbf{A}_{n-3}$ con el vector $\mathbf{X}$ igual a la $(n-2)$ columna de la matriz $\mathbf{A_{n-3}}$. En forma general, $\mathbf{P_{n-2}}$ es

$$\mathbf{P_{n-2}} = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & & & \vdots \\ 0 & 0 & & 1 & & 0 \\ \vdots & \vdots & & & p_{n-1,\,n-1} & p_{n-1,\,n} \\ 0 & 0 & \cdots & 0 & p_{n,\,n-1} & p_{n,\,n} \end{bmatrix}$$

El bloque de identidad de $((n-2)\times(n-2))$ en la parte superior izquierda garantiza que la tridiagonalización lograda en el paso anterior no se altere por la siguiente transformación, $\mathbf{A_{n-2}} = \mathbf{P_{n-2}}\mathbf{A_{n-3}}\mathbf{P_{n-2}}$. La matriz tridiagonal resultante tiene la forma siguiente:

www.FreeLibros.me

$$\mathbf{A_{n-2}} = \mathbf{P_{n-2}}\mathbf{A_{n-3}}\mathbf{P_{n-2}} = \begin{bmatrix} a_{11} & u_1 & 0 & 0 & \cdots & 0 \\ u_1 & z_1 & u_2 & 0 & \cdots & 0 \\ 0 & u_2 & z_2 & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & \ddots & u_{n-2} & 0 \\ \vdots & \vdots & \ddots & u_{n-2} & z_{n-2} & w_{n-1,n} \\ 0 & 0 & \cdots & 0 & w_{n,n-1} & w_{n,n} \end{bmatrix}$$

El vector $\mathbf{W}^{(k)}$ se forma de la siguiente manera

$$\left[\mathbf{W}^{(k)}\right]^{\mathrm{T}} = \frac{1}{R}\left[0 \quad \cdots \quad 0 \quad (x_{k+1}+S) \quad x_{k+2} \quad \cdots \quad x_n\right]$$

donde

k es el número de transformación.

$$S = sign(x_{k+1})\sqrt{\sum_{j=k+1}^{n}(x_j)^2}$$

$$R = \sqrt{2(S)(x_{k+1}+S)}$$

La sección 8.9.1 proporciona el código desarrollado en Matlab para reducir una matriz simétrica a una forma tridiagonal utilizando el método de Householder.

EJEMPLO 8.2

Usar el método de Householder para reducir a una matriz a su forma tridiagonal simétrica,

$$\mathbf{A} = \begin{bmatrix} \mathbf{8} & 2 & 5 & 2 & 7 & 2 \\ 2 & 5 & 4 & 7 & 2 & \mathbf{8} \\ 5 & 4 & 7 & 3 & 5 & 4 \\ 2 & 7 & 3 & 3 & 7 & 9 \\ 7 & 2 & 5 & 7 & 4 & 1 \\ 2 & \mathbf{8} & 4 & 9 & 1 & 9 \end{bmatrix}$$

Se calcula la primera matriz de transformación donde $k=1$

$$\left[\mathbf{X}^{(1)}\right]^{\mathrm{T}} = [8 \quad 2 \quad 5 \quad 2 \quad 7 \quad 2]$$

$$S = (1)\sqrt{(2)^2+(5)^2+(2)^2+(7)^2+(2)^2} = 9.2736$$

$$R = \sqrt{2(9.2736)(2+9.2736)} = 14.4601$$

$$\mathbf{W}^{(1)} = \frac{1}{14.4601}\begin{bmatrix} 0 \\ 2+9.2736 \\ 5 \\ 2 \\ 7 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.7796 \\ 0.3458 \\ 0.1383 \\ 0.4841 \\ 0.1383 \end{bmatrix}$$

www.FreeLibros.me

$$\mathbf{P}^{(1)}=\mathbf{I}-2\mathbf{W}^{(1)}\left[\mathbf{W}^{(1)}\right]^{T}=\begin{bmatrix}1&0&0&0&0&0\\0&-0.2157&-0.5392&-0.2157&-0.7548&-0.2157\\0&-0.5392&0.7609&-0.0957&-0.3348&-0.0957\\0&-0.2157&-0.0957&0.9617&-0.1339&-0.0383\\0&-0.7548&-0.3348&-0.1339&0.5313&-0.1339\\0&-0.2157&-0.0957&-0.0383&-0.1339&0.9617\end{bmatrix}$$

$$\mathbf{A}^{(1)}=\mathbf{P}^{(1)}\mathbf{A}\mathbf{P}^{(1)}=\begin{bmatrix}8&-9.2736&0&0&0&0\\-9.2736&17.2209&1.2562&-6.5282&7.7757&-4.0481\\0&1.2562&2.1622&-4.4483&2.4924&-2.7918\\0&-6.5282&-4.4483&-2.1846&-1.7215&4.0780\\0&7.7757&2.4924&-1.7215&6.4608&-6.8024\\0&-4.0481&-2.7918&4.0780&-6.8024&4.3406\end{bmatrix}$$

Se calcula la segunda matriz de transformación donde $k=2$

$$\left[\mathbf{X}^{(2)}\right]^{T}=\left[-9.2736\quad 17.2209\quad 1.2562\quad -6.5282\quad 7.7757\quad -4.0481\right]$$

$$S=(1)\sqrt{(1.2562)^{2}+(-6.5282)^{2}+(7.7757)^{2}+(-4.0481)^{2}}=11.0020$$

$$R=\sqrt{2(11.0020)(1.2562+11.0020)}=16.4234$$

$$\mathbf{W}^{(2)}=\frac{1}{16.4234}\begin{bmatrix}0\\0\\1.2562+11.0020\\-6.5282\\7.7757\\-4.0481\end{bmatrix}=\begin{bmatrix}0\\0\\0.7464\\-0.3975\\0.4735\\-0.2465\end{bmatrix}$$

$$\mathbf{P}^{(2)}=\mathbf{I}-2\mathbf{W}^{(2)}\left[\mathbf{W}^{(2)}\right]^{T}=\begin{bmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&-0.1142&0.5934&-0.7068&0.3679\\0&0&0.5934&0.6840&0.3764&-0.1960\\0&0&-0.7068&0.3764&0.5517&0.2334\\0&0&0.3679&-0.1960&0.2334&0.8785\end{bmatrix}$$

$$\mathbf{A}^{(2)}=\mathbf{P}^{(2)}\mathbf{A}^{(1)}\mathbf{P}^{(2)}=\begin{bmatrix}8&-9.2736&0&0&0&0\\-9.2736&17.2209&-11.0020&0&0&0\\0&-11.0020&11.0757&-6.9921&2.2504&3.6115\\0&0&-6.9921&-2.0032&-0.1950&-0.0597\\0&0&2.2504&-0.1950&2.5672&-0.7935\\0&0&3.6115&-0.0597&-0.7935&-0.8607\end{bmatrix}$$

www.FreeLibros.me

Se calcula la tercera matriz de transformación donde $k=3$

$$\left[\mathbf{X}^{(3)}\right]^{\mathrm{T}}=\begin{bmatrix}0 & -11.0020 & 11.0757 & -6.9921 & 2.2504 & 3.6115\end{bmatrix}$$

$$S=(-1)\sqrt{(-6.9921)^2+(2.2504)^2+(3.6115)^2}=-8.1851$$

$$R=\sqrt{2(-8.1651)(-6.9921-8.1651)}=15.7625$$

$$\mathbf{W}^{(3)}=\frac{1}{15.7625}\begin{bmatrix}0\\0\\0\\-6.9921-8.1851\\2.2504\\3.6115\end{bmatrix}=\begin{bmatrix}0\\0\\0\\-0.9629\\0.1428\\0.2291\end{bmatrix}$$

$$\mathbf{P}^{(3)}=\mathbf{I}-2\mathbf{W}^{(3)}\left[\mathbf{W}^{(3)}\right]^{\mathrm{T}}=\begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0\\0 & 1 & 0 & 0 & 0 & 0\\0 & 0 & 1 & 0 & 0 & 0\\0 & 0 & 0 & -0.8542 & 0.2749 & 0.4412\\0 & 0 & 0 & 0.2749 & 0.9592 & -0.0654\\0 & 0 & 0 & 0.4412 & -0.0654 & 0.8950\end{bmatrix}$$

$$\mathbf{A}^{(3)}=\mathbf{P}^{(3)}\mathbf{A}^{(2)}\mathbf{P}^{(3)}=\begin{bmatrix}8 & -9.2736 & 0 & 0 & 0 & 0\\-9.2736 & 17.2209 & -11.0020 & 0 & 0 & 0\\0 & -11.0020 & 11.0757 & 8.1851 & 0 & 0\\0 & 0 & 8.1851 & -1.4912 & 0.9853 & 0.1961\\0 & 0 & 0 & 0.9853 & 2.2059 & -1.1304\\0 & 0 & 0 & 0.1961 & -1.1304 & -1.0114\end{bmatrix}$$

Se calcula la cuarta matriz de transformación (última para este caso) donde $k=4=n-2$

$$\left[\mathbf{X}^{(4)}\right]^{\mathrm{T}}=\begin{bmatrix}0 & 0 & 8.1851 & -1.4912 & 0.9853 & 0.1961\end{bmatrix}$$

$$S=(1)\sqrt{(0.9853)^2+(0.1961)^2}=1.0046$$

$$R=\sqrt{2(1.0046)(0.9853+1.0046)}=1.9995$$

$$\mathbf{W}^{(4)}=\frac{1}{1.9995}\begin{bmatrix}0\\0\\0\\0\\0.9853+1.0046\\0.1961\end{bmatrix}=\begin{bmatrix}0\\0\\0\\0\\0.9952\\0.0981\end{bmatrix}$$

www.FreeLibros.me

$$\mathbf{P}^{(4)} = \mathbf{I} - 2\mathbf{W}^{(4)}\left[\mathbf{W}^{(4)}\right]^{\mathrm{T}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.9808 & -0.1952 \\ 0 & 0 & 0 & 0 & -0.1952 & 0.9808 \end{bmatrix}$$

$$\mathbf{A}^{(4)} = \mathbf{P}^{(4)}\mathbf{A}^{(3)}\mathbf{P}^{(4)} = \begin{bmatrix} 8 & -9.2736 & 0 & 0 & 0 & 0 \\ -9.2736 & 17.2209 & -11.0020 & 0 & 0 & 0 \\ 0 & -11.0020 & 11.0757 & 8.1851 & 0 & 0 \\ 0 & 0 & 8.1851 & -1.4912 & -1.0046 & 0 \\ 0 & 0 & 0 & -1.0046 & 1.6505 & 1.6602 \\ 0 & 0 & 0 & 0 & 1.6602 & -0.4560 \end{bmatrix}$$

De esta forma se llega a una matriz tridiagonal simétrica, la cual es similar a la matriz original.

8.7.4 Multiplicación sucesiva por $y^{(k)}$

Los métodos iterativos son más adecuados para problemas en los que se necesitan encontrar sólo una o dos raíces, aunque hay forma de extender estos métodos para encontrar cualquier cantidad de raíces. Aquí se supone que todos los valores propios son distintos. Por tanto, los vectores propios son linealmente independientes. Así, un vector arbitrario se puede expresar en la forma

$$\mathbf{y} = \sum_{r=1}^{n} a_r \mathbf{X}^{(\mathbf{r})} \tag{8.64}$$

Para encontrar el valor propio más grande y su correspondiente vector propio, se multiplica un vector arbitrario $\mathbf{y}^{(\mathbf{0})}$, sucesivamente, por la matriz $\mathbf{A}$. En general, la secuencia de vectores $\mathbf{y}^{(\mathbf{k})}$ va a converger al vector propio, y la proporción de elementos sucesivos va a converger al valor propio. Por ejemplo, si se deja que λ_1 sea el valor propio más grande y $\mathbf{X}^{(\mathbf{1})}$ su vector propio correspondiente. entonces

$$\mathbf{y}^{(\mathbf{k})} = \mathbf{A}^{(\mathbf{k})}\mathbf{y}^{(\mathbf{0})} = \mathbf{A}^{(\mathbf{k})}\sum_{r=1}^{n} a_r \mathbf{X}^{(\mathbf{r})} = \mathbf{A}^{(\mathbf{k})}\sum_{r=1}^{n} a_r \lambda_i^{\,k} \mathbf{X}^{(\mathbf{r})} \tag{8.65}$$

Así,

$$\lambda_i^{\,k} = \left[a_1\mathbf{X}^{(\mathbf{1})} + a_2\left(\frac{\lambda_2}{\lambda_1}\right)^{\mathbf{k}} \mathbf{X}^{(2)} + \cdots + \left(\frac{\lambda_n}{\lambda_1}\right)^{\mathbf{k}} \mathbf{X}^{(\mathbf{n})} \right] \tag{8.66}$$

El valor de $\left(\dfrac{\lambda_r}{\lambda_1}\right)^k (r \neq 0)$ tiende a cero cuando k tiende a ∞ y así, todos los términos son despreciables, excepto el primero de ellos. Entonces, $\mathbf{y}^{(\mathbf{k})}$ tiende a un escalar múltiplo de $\mathbf{X}^{(\mathbf{1})}$, y la relación de un elemento $y_i^{(k+1)}$, para el elemento correspondiente $y_i^{(k)}$, tiende a λ_1 cuando se incrementa k.

Se puede observar que la convergencia será más rápida si el vector inicial contiene una gran componente de $\mathbf{X}^{(\mathbf{1})}$. Si las componentes de $\mathbf{X}^{(\mathbf{1})}$ son completamente erróneas, por ejemplo, $a_1 = 0$, entonces parece que la secuencia no puede converger al vector propio dominante.

Sin embargo, los errores de redondeo normalmente van a producir una componente de $\mathbf{X}^{(\mathbf{1})}$ amplificada que, finalmente, será la dominante. Si la convergencia se hace lenta con la elección de un vector en particular, entonces algunas veces se puede hacer un progreso más satisfactorio eligiendo un vector inicial

www.FreeLibros.me

diferente. La rapidez de la convergencia se ve claramente afectada por la proporción del módulo de los dos valores propios más grandes. Cuando esta proporción está cerca de la unidad, da como resultado una convergencia muy lenta.

El proceso computacional es ligeramente diferente del proceso descrito anteriormente, ya que el crecimiento ilimitado de $\mathbf{y}^{(\mathbf{k})}$ se puede evitar de la siguiente manera:

1. El vector $\mathbf{y}^{(0)}$ se normaliza de acuerdo con el elemento de módulo más grande.
2. El vector se multiplica por la matriz **A**.
3. El nuevo vector se normaliza dividiendo cada elemento entre el elemento de módulo más grande, el cual designamos como q_k.
4. El vector $\mathbf{y}^{(\mathbf{k})}$ se multiplica repetidamente por la matriz **A** y se divide entre el factor q_k hasta que el valor de q_k y q_{k+1} difieran en un valor pequeño previamente especificado. El valor de q_k da el valor propio más grande y el vector $\mathbf{y}^{(\mathbf{k})}$ es el vector propio correspondiente.

La sección 8.9.2 proporciona el código desarrollado en Matlab para determinar los valores y vectores propios de una matriz de 3×3 utilizando la técnica de multiplicación sucesiva por un vector cualesquiera.

EJEMPLO 8.3

Utilizando el método de multiplicación sucesiva, calcular el valor propio de mayor módulo y su vector propio correspondiente, de la siguiente matriz:

$$\mathbf{A}=\begin{bmatrix}7 & 2 & 3\\ 6 & 5 & 4\\ 2 & 8 & 9\end{bmatrix}$$

La expresión a iterar tiene la estructura $\mathbf{Y}^{(k+1)}=\mathbf{A}\mathbf{Y}^{(k)}$. Por tanto, si se tiene un vector inicial cualquiera como $\left[\mathbf{Y}^{(0)}\right]^T=[1 \quad 4 \quad 8]$, como primera iteración después de normalizar el vector $\mathbf{Y}^{(k)}$ en forma lineal, es decir dividiendo todo el vector entre el elemento de mayor módulo para hacer el número más grande unitario, se tiene

$$\mathbf{Y}^{(1)}=\mathbf{A}\mathbf{Y}^{(0)}=\begin{bmatrix}7 & 2 & 3\\ 6 & 5 & 4\\ 2 & 8 & 9\end{bmatrix}\begin{bmatrix}0.1250\\ 0.5000\\ 1.0000\end{bmatrix}$$

En la siguiente tabla se resumen las siguientes iteraciones hasta encontrar dos soluciones consecutivas con un error de una milésima.

Tabla 8.1 Resultados de aplicar el método de multiplicación sucesiva de una matriz, por un vector para encontrar el valor propio de módulo más grande y su vector propio.

#iteración	0	1	2	3	4	5	6	7	8	9
$\mathbf{Y}^{(k)}$	0.1250 0.5000 1.0000	0.3679 0.5472 1.0000	0.4726 0.6337 1.0000	0.5045 0.6663 1.0000	0.5127 0.6753 1.0000	0.5146 0.6775 1.0000	0.5151 0.6780 1.0000	0.5152 0.6782 1.0000	0.5152 0.6782 1.0000	0.5152 0.6782 1.0000
q_k	8.0000	13.2500	14.1132	15.0147	15.3393	15.4278	15.4494	15.4544	15.4556	15.4558

www.FreeLibros.me

8.7.4.1 Raíces complejas conjugadas

Si la raíz de módulo más grande es compleja, entonces, ya que la matriz **A** es real, debe haber otra raíz de igual módulo, que es el complejo conjugado. El análisis previo no es muy relevante, y las multiplicaciones por la matriz **A** no van a producir una secuencia de vectores que converja. Si el vector inicial arbitrario $\mathbf{y}^{(0)}$ se expande en términos de sus valores propios, entonces, después de muchas multiplicaciones por la matriz A, todos los términos, excepto dos serán insignificantes. Así,

$$\mathbf{y}^{(\mathbf{k})} \approx a_1\lambda_1^k\mathbf{X}^{(1)} + a_2\lambda_1^k\overline{\mathbf{X}}^{(1)} \tag{8.67}$$

El siguiente análisis muestra cómo se pueden encontrar los valores propios. Si se deja que λ_1 y $\overline{\lambda}_1$ sean las soluciones de la ecuación cuadrática

$$\lambda^2 + a\overline{\lambda} + b = 0, \tag{8.68}$$

donde a y b son cantidades desconocidas en esta etapa, se toman tres vectores consecutivos de la secuencia $\mathbf{y}^{(\mathbf{k})}$, $\mathbf{y}^{(\mathbf{k+1})}$ y $\mathbf{y}^{(\mathbf{k+2})}$, y se forma la siguiente combinación lineal

$$\begin{aligned}\mathbf{y}^{(\mathbf{k+2})} + a\mathbf{y}^{(\mathbf{k+1})} + b\mathbf{y}^{(\mathbf{k})} &\approx a_1\lambda_1^{k+2}\mathbf{X}^{(1)} + a_1\overline{\lambda}_1^{k+2}\overline{\mathbf{X}}^{(1)} + a\left[a_1\lambda_1^{k+1}\mathbf{X}^{(1)} + a_1\overline{\lambda}_1^{k+1}\overline{\mathbf{X}}^{(1)}\right]\\ &+ b\left[a_1\lambda_1^k\mathbf{X}^{(1)} + a_1\overline{\lambda}_1^k\overline{\mathbf{X}}^{(1)}\right]\end{aligned} \tag{8.69}$$

o bien,

$$\mathbf{y}^{(\mathbf{k+2})} + a\mathbf{y}^{(\mathbf{k+1})} + b\mathbf{y}^{(\mathbf{k})} \approx a_1\lambda_1^k\mathbf{X}^{(1)}\left[\lambda_1^2 + a\lambda_1 + b\right] + a_2\overline{\lambda}_1^k\overline{\mathbf{X}}^{(1)}\left[\overline{\lambda}_1^2 + a\overline{\lambda}_1 + b\right], \tag{8.70}$$

entonces, conociendo a y b, se puede formar una combinación lineal de tres vectores consecutivos que será, aproximadamente, igual al vector nulo. El problema considerado es el cálculo de la inversa. Por tanto, se trata de encontrar valores de a y b tales que una combinación lineal de tres vectores consecutivos den el vector nulo. Si se toman dos componentes cualesquiera de estos vectores y la combinación lineal se iguala a cero, se generan dos ecuaciones para los dos valores desconocidos a y b. Esto normalmente se puede resolver como

$$\mathbf{y}_\mathbf{r}^{\mathbf{k+2}} + a\mathbf{y}_\mathbf{r}^{\mathbf{k+1}} + b\mathbf{y}_\mathbf{r}^{\mathbf{k}} = 0 \tag{8.71a}$$

$$\mathbf{y}_\mathbf{s}^{\mathbf{k+2}} + a\mathbf{y}_\mathbf{s}^{\mathbf{k+1}} + b\mathbf{y}_\mathbf{s}^{\mathbf{k}} = 0 \tag{8.71b}$$

Una vez que se conocen los valores de a y b, entonces la ecuación (8.68) se puede resolver para encontrar los valores de λ_1 y $\overline{\lambda}_1$. El vector propio $\mathbf{X}^{(1)}$ se encuentra a partir de dos vectores consecutivos

$$\mathbf{y}^{(\mathbf{k+1})} - \overline{\lambda}_1\mathbf{y}^{(\mathbf{k})} = a_1\lambda_1^k\left[\lambda_1 - \overline{\lambda}_1\right]\mathbf{X}^{(1)} \tag{8.72}$$

y $\overline{\mathbf{X}}^{(1)}$ se encuentra de una forma similar

$$\mathbf{y}^{(\mathbf{k+1})} - \lambda_1\mathbf{y}^{(\mathbf{k})} = a_1\overline{\lambda}_1^k\left[\overline{\lambda}_1 - \lambda_1\right]\overline{\mathbf{X}}^{(1)} \tag{8.73}$$

Prácticamente, el proceso iterativo continúa hasta que los valores de a y b sean en efecto constantes, sin importar cuáles componentes de los vectores $\mathbf{y}^{(\mathbf{k})}$ se usen, y tampoco cambian los valores de una iteración a otra.

8.7.4.2 Iteración inversa para la raíz más pequeña

Con una simple modificación al método anterior, es posible usar iteraciones para encontrar el valor propio más pequeño. La penalidad es que el nuevo proceso involucra la solución repetida de conjuntos de ecuaciones lineales simultáneas. La solución se basa en la propiedad de que los valores propios de $\mathbf{A}^{-1}$ son los inversos de los valores propios de **A**; por tanto, el valor propio más pequeño de **A** es el valor propio

www.FreeLibros.me

más grande de $\mathbf{A}^{-1}$. Los vectores propios de $\mathbf{A}$ y de $\mathbf{A}^{-1}$ son los mismos. Por tanto, se itera con $\mathbf{A}^{-1}$, ejemplo:

$$\mathbf{y}^{(\mathbf{k})} = \mathbf{A}^{-1}\mathbf{y}^{(\mathbf{k-1})} \tag{8.74}$$

Sin embargo, se ha mostrado que el proceso de encontrar $\mathbf{A}^{-1}$ de modo explícito es ineficiente y se puede reemplazar por la eliminación gaussiana. Esto se puede hacer en este problema encontrando la secuencia de vectores $\mathbf{y}^{(\mathbf{k})}$ resolviendo la sucesión de ecuaciones

$$\mathbf{y}^{(\mathbf{k})} = \mathbf{A}^{-1}\mathbf{y}^{(\mathbf{k-1})}, \qquad k = 1, 2, \ldots \tag{8.75}$$

Después de la primera reducción a la forma triangular, la solución que le sigue utiliza la multiplicación por una matriz triangular y la sustitución regresiva, de manera que se reduzca el tiempo de cálculo. Después de cada solución, el vector $\mathbf{y}^{(\mathbf{k})}$ se normaliza dividiendo entre el elemento de mayor módulo q_k. El proceso se detiene cuando la diferencia entre dos valores sucesivos de q_k es menor que algún valor especificado. La magnitud del valor propio más pequeño de $\mathbf{A}$ está dado por q_k^{-1}, y el vector propio por $\mathbf{y}^{(\mathbf{k})}$.

8.7.4.3 Encontrar la raíz más cercana al valor dado

Si $\mathbf{B} = \mathbf{A} - p\mathbf{I}$, donde se pide el valor propio más cercano a p, los vectores propios de $\mathbf{A}$ satisfacen la ecuación $\mathbf{AX} = \lambda\mathbf{X}$ y, por tanto, también satisfacen la ecuación

$$(\mathbf{A} - p\mathbf{I})\mathbf{X} = (\lambda - p)\mathbf{X} \tag{8.76}$$

Esto significa que los vectores propios de $\mathbf{A}$ y de $\mathbf{B}$ son los mismos, y los nuevos valores propios son $(\lambda - p)$. Así, el valor propio λ_i más cercano a p corresponde al valor propio más pequeño de $\mathbf{B}$, y el método de la sección previa se puede usar para encontrar $(\lambda_i - p)^{-1}$. Si el divisor es q_k, como antes, entonces $\lambda_i = \dfrac{p+1}{q_k}$, y el vector propio es igual a $\mathbf{y}^{(\mathbf{k})}$.

8.7.4.4 Extensión del método

En seguida se describe un proceso para extender el esquema anterior para encontrar todos los valores propios de una matriz. Sin embargo, el proceso no se recomienda para matrices muy grandes, ya que el error que se va acumulando puede ser al final muy grande. La base de este método es eliminar, de alguna manera, el valor propio más grande y su respectivo vector propio, de tal forma que el esquema iterativo anterior seleccione el siguiente valor propio más grande. El proceso se conoce como *deflación*. Si se deja que λ_1 sea el valor propio más grande y que $\mathbf{X}^{(1)}$ sea su vector propio correspondiente, la matriz $\mathbf{A}$ se divide suponiendo que el elemento de $\mathbf{X}^{(1)}$ es el más grande. Si se supone que el primer elemento es el más grande, la matriz $\mathbf{A}$ se representa por

$$\mathbf{A} = \begin{pmatrix} \mathbf{a_1} \\ \mathbf{B} \end{pmatrix} \tag{8.77}$$

donde $\mathbf{a_1}$ es el primer renglón de $\mathbf{A}$, y $\mathbf{B}$ es la matriz de $[(n-1)\times n]$ de los renglones que quedan. Debido a que el componente más grande del vector $\mathbf{X}^{(1)}$ es el primero, todos los vectores afectados se normalizan de tal manera que el primer elemento sea unitario. Se forma una matriz de la forma

$$\mathbf{A_1} = \mathbf{A} - \mathbf{X}^{(1)}\mathbf{a_1} \tag{8.78}$$

Se va a demostrar que $\mathbf{A_1}$ tiene valores propios $\lambda_i \;\; (i = 1, 2, \ldots, n)$ que son los mismos que para $\mathbf{A}$, y los valores propios restantes son iguales a cero. Considere cualquier otro valor propio λ_i de $\mathbf{A}$ con su correspondiente vector propio $\mathbf{X}^{(\mathbf{i})}$,

$$\mathbf{A_1}\left(\mathbf{X}^{(1)} - \mathbf{X}^{(\mathbf{i})}\right) = \mathbf{A}\left(\mathbf{X}^{(1)} - \mathbf{X}^{(\mathbf{i})}\right) - \mathbf{X}^{(1)}\mathbf{a_1}\left(\mathbf{X}^{(1)} - \mathbf{X}^{(\mathbf{i})}\right) \tag{8.79}$$

www.FreeLibros.me

Debido a que $\mathbf{a_1}$ es el primer renglón de la matriz $\mathbf{A}$, el producto $\mathbf{a_1X^{(i)}}$ es el primer elemento del vector $\lambda_i\mathbf{X^{(i)}}$. Esto es igual a λ_i debido a que el vector está normalizado.

Así, el lado derecho de la ecuación (8.79) es igual a

$$\lambda_1\mathbf{X^{(1)}}-\lambda_i\mathbf{X^{(i)}}-\mathbf{X^{(1)}}(\lambda_1-\lambda_i)=\lambda_i\left(\mathbf{X^{(1)}}-\mathbf{X^{(i)}}\right) \tag{8.80}$$

Esto muestra que los valores propios $\lambda_i\ \ (i=1,2,\ldots,n)$ son valores propios de la nueva matriz, y los vectores propios correspondientes son $\left(\mathbf{X^{(1)}}-\mathbf{X^{(i)}}\right)$. También

$$\mathbf{A_1X^{(1)}}=\mathbf{AX^{(1)}}-\mathbf{X^{(1)}a_1X^{(1)}}=\mathbf{AX^{(1)}}-\lambda_1\mathbf{X^{(1)}}=0, \tag{8.81}$$

de tal forma que el valor propio restante es cero. La matriz $\mathbf{A_1}$ ahora se puede usar para iterar para el próximo valor propio más grande, y así sucesivamente.

Cuando se ha encontrado el vector propio $\mathbf{r^{(2)}}$, correspondiente al valor propio más grande de la matriz $\mathbf{A_1}$, se puede usar para encontrar el vector propio de $\mathbf{A}$, correspondiente al valor propio λ_2. Se supone que λ_2 es el próximo valor propio más grande. De lo anterior se tiene que $\mathbf{r^{(2)}}$ es un escalar múltiplo de $\left(\mathbf{X^{(1)}}-\mathbf{X^{(2)}}\right)$, por ejemplo

$$c\mathbf{r^{(2)}}=\mathbf{X^{(1)}}-\mathbf{X^{(2)}} \tag{8.82}$$

Así,

$$\mathbf{X^{(2)}}=\mathbf{X^{(1)}}-c\mathbf{r^{(2)}} \tag{8.83}$$

donde c es la constante que se debe encontrar. También,

$$\mathbf{a_1X^{(2)}}=\mathbf{a_1X^{(1)}}-c\mathbf{a_1r^{(2)}} \tag{8.84}$$

$$\lambda_2=\lambda_1-c\,\mathbf{a_1r^{(2)}} \tag{8.85}$$

y de aquí se determina el valor de c,

$$c=\frac{(\lambda_1-\lambda_2)}{\mathbf{a_1r^{(2)}}} \tag{8.86}$$

que entonces se puede sustituir en la ecuación (8.83) para dar $\mathbf{X^{(2)}}$.

8.7.5 Método de potenciación

Este método usa la característica de una matriz que sea diagonalizable; es decir, se puede encontrar una matriz de la forma

$$\mathbf{M^{-1}AM}=\Lambda \tag{8.87}$$

donde $\mathbf{M}$ es la llamada *matriz de vectores propios* de $\mathbf{A}$ que la "diagonalizan". Esto conduce a

$$\mathbf{A}=\mathbf{M}\Lambda\mathbf{M^{-1}} \tag{8.88}$$

Si se asocia $\mathbf{C}=\mathbf{M}$ y $\mathbf{R}=\mathbf{M^{-1}}$, se obtiene

$$\mathbf{A}=\mathbf{C}\Lambda\mathbf{R} \tag{8.89}$$

Expresando la ecuación anterior en particiones de $\mathbf{C}$ y $\mathbf{R}$, se puede representar como

$$\mathbf{A}=\mathbf{C}_1\lambda_1\mathbf{R}_1+\cdots+\mathbf{C}_n\lambda_n\mathbf{R}_n \tag{8.90}$$

www.FreeLibros.me

Como λ es un valor constante, se puede conmutar para dar

$$\mathbf{A} = \lambda_1 \mathbf{C}_1 \mathbf{R}_1 + \cdots + \lambda_n \mathbf{C}_n \mathbf{R}_n \tag{8.91}$$

donde $\mathbf{C}_n \mathbf{R}_n$ se asocia a la n-ésima *matriz idempotente*, es decir:

$$\mathbf{I}_n = \mathbf{C}_n \mathbf{R}_n \tag{8.92}$$

Así, el proceso finalmente conduce a

$$\mathbf{A} = \lambda_1 \mathbf{I}_1 + \cdots + \lambda_n \mathbf{I}_n \tag{8.93}$$

Este método se basa en el hecho de que las matrices idempotentes son ortogonales. Esto significa que la multiplicación entre dos matrices idempotentes diferentes es cero, y la multiplicación de una matriz idempotente por sí misma es diferente de cero. Las matrices idempotentes también son ortonormales; es decir, si una matriz idempotente se eleva a una potencia, como resultado se tiene la misma matriz idempotente. Estas dos características conducen al siguiente resultado:

$$\mathbf{A}^2 = \lambda_1^2 \mathbf{I}_1 + \cdots + \lambda_n^2 \mathbf{I}_n \tag{8.94}$$

Si se repite este mismo proceso un gran número de veces, entonces el valor propio de mayor módulo va a dominar para finalmente obtener

$$\mathbf{A}^n \cong \lambda_x^n \mathbf{I}_x \tag{8.95}$$

donde λ_x es el vector propio dominante y $\mathbf{I}_x$ es su matriz idempotente asociada. Así, la repetición de $\mathbf{A}^2$ va a converger a una matriz proporcional de la matriz idempotente $\mathbf{I}_x$.

Debido al hecho de desbordamiento numérico, en cada etapa el resultado de la potenciación se normaliza. Este proceso converge cuando dos matrices normalizadas consecutivas son iguales, o su diferencia es menor a un valor especificado.

El coeficiente de proporcionalidad para obtener la matriz idempotente se calcula como la razón de cambio del último paso $\mathbf{A}^n$ y su potenciación. Si $\mathbf{B} = \mathbf{A}^n$, se obtiene este coeficiente como

$$k = \mathbf{B} / \mathbf{B}^2 \tag{8.96}$$

Así, la matriz idempotente es

$$\mathbf{I}_x = k \cdot \mathbf{B} \tag{8.97}$$

Las columnas de la matriz idempotente $\mathbf{I}_x$ son proporcionales al vector propio asociado al valor propio λ_x. Por tanto, el valor propio se calcula utilizando una de las columnas de la matriz idempotente y la relación

$$\mathbf{A} * \mathbf{I}_x = \lambda_x \cdot \mathbf{I}_x \tag{8.98}$$

El próximo paso es sustraer la matriz idempotente encontrada de la matriz original, esto conduce a

$$\mathbf{A}_1 = \mathbf{A} - \lambda_1 \mathbf{I}_1 = \lambda_2 \mathbf{I}_2 + \cdots + \lambda_n \mathbf{I}_n \tag{8.99}$$

La matriz resultante se usa ahora para calcular la siguiente idempotente asociada al próximo valor propio, el cual será el de mayor módulo de entre los restantes. Este proceso se repite hasta que se encuentra la última matriz idempotente y su valor propio asociado.

Este método tiene dos pequeñas desventajas que se pueden superar con cierta facilidad:

a) La primera se refiere al hecho de tener valores propios complejos conjugados cuando se tiene una matriz puramente real. El método en este caso no converge, ya que la potenciación de una matriz

www.FreeLibros.me

real siempre da una matriz real. De esta forma es imposible obtener un número complejo de la potenciación; sin embargo, basta sumarle un pequeño valor imaginario a alguno de los coeficientes de la matriz para que ésta converja. El error al agregar este pequeño valor es mínimo y, hasta cierto punto, se puede decir que es menor al error propio que introduce la computadora por redondeo.

b) El segundo problema se refiere al hecho de tener una matriz con valores propios repetidos. Esto se debe a que converge a una matriz idempotente que es, a su vez, la suma de dos matrices idempotentes. El problema consiste en que la separación de ambas es muy compleja; sin embargo, para el caso de valores propios repetidos se usa la técnica de la forma canónica de la matriz descrita en el apartado 8.3.

La sección 8.9.3 proporciona el código desarrollado en Matlab para el cálculo de valores y vectores propios de una matriz cualesquiera utilizando el método de potenciación.

EJEMPLO 8.4

Utilizando el método de potenciación, determinar las matrices idempotentes, los valores propios y los vectores propios de la matriz del ejemplo anterior,

$$\mathbf{A} = \begin{bmatrix} 7 & 2 & 3 \\ 6 & 5 & 4 \\ 2 & 8 & 9 \end{bmatrix}$$

Después de potenciar la matriz se determina la primera idempotente (las columnas de esta matriz son proporcionales al vector propio) y su correspondiente valor propio como

$$\mathbf{Idem}_1 = \begin{bmatrix} 0.2212 & 0.2300 & 0.2453 \\ 0.2912 & 0.3027 & 0.3229 \\ 0.4293 & 0.4464 & 0.4761 \end{bmatrix}$$

$$\lambda_1 = 15.4559$$

La matriz que resulta de la operación $\mathbf{B} = \mathbf{A} - \lambda_1 \mathbf{Idem}_1$ se utiliza para determinar la siguiente matriz idempotente y el valor propio asociado. Por tanto, se tiene

$$\mathbf{B} = \begin{bmatrix} 3.5813 & -1.5544 & -0.7909 \\ 1.4997 & 0.3210 & -0.9903 \\ -4.6358 & 1.1008 & 1.6418 \end{bmatrix}$$

$$\mathbf{Idem}_2 = \begin{bmatrix} 1.5538 & -0.8655 & -0.2135 \\ 1.7427 & -0.9708 & -0.2394 \\ -3.0352 & 1.6908 & 0.4170 \end{bmatrix}$$

$$\lambda_2 = 3.3829$$

La matriz que resulta de la operación $\mathbf{C} = \mathbf{B} - \lambda_2 \mathbf{Idem}_2$ se utiliza para determinar la siguiente matriz idempotente y el valor propio asociado. Por tanto,

$$\mathbf{C} = \begin{bmatrix} -1.6749 & 1.3736 & -0.0687 \\ -4.3957 & 3.6050 & -0.1803 \\ 5.6318 & -4.6189 & 0.2310 \end{bmatrix}$$

www.FreeLibros.me

$$\mathbf{Idem}_2 = \begin{bmatrix} -0.7750 & 0.6356 & -0.0318 \\ -2.0339 & 1.6681 & -0.0834 \\ 2.6059 & -2.1372 & 0.1069 \end{bmatrix}$$

$$\lambda_3 = 2.1612$$

La comprobación de que la descomposición en matrices idempotentes está bien, se puede realizar con la ecuación (8.93).

8.7.6 Métodos L-R y Q-R

Existen dos métodos que se pueden usar para encontrar todos los valores propios de una matriz real o compleja. Estos métodos son los más eficientes cuando se necesita encontrar todos los valores propios de una matriz. El más sencillo de estos métodos es el algoritmo **L-R**, cuyo nombre se debe al proceso de cómputo por la factorización repetida de una secuencia de matrices, de la forma triangular izquierda y triangular derecha. Para ajustarse con la notación previa, la notación de letras **L** y **U** se usan para matrices de forma triangular inferior y triangular superior, respectivamente.

Al principio del proceso se forma la sucesión de matrices por descomposición triangular de cada miembro de la secuencia. Se supone, para propósitos de procedimiento, que todas las matrices son de tal forma que la descomposición triangular es posible. Si se deja que

$$\mathbf{A} = \mathbf{A}_1 = \mathbf{L}_1\mathbf{U}_1 \tag{8.100}$$

y se forma

$$\mathbf{A}_2 = \mathbf{U}_1\mathbf{L}_1 = \mathbf{L}_2\mathbf{U}_2 \tag{8.101}$$

o bien, generalizando,

$$\mathbf{A}_\mathbf{r} = \mathbf{U}_{\mathbf{r}-1}\mathbf{L}_{\mathbf{r}-1} = \mathbf{L}_\mathbf{r}\mathbf{U}_\mathbf{r}, \qquad r = 1, 2, \ldots \tag{8.102}$$

Se puede observar que estas matrices son *similares* a $\mathbf{A}_1$ y, por tanto, tienen los mismos valores propios, debido a que

$$\mathbf{A}_2 = (\mathbf{U}_1)\mathbf{L}_1 = \mathbf{L}_1^{-1}\mathbf{A}_1\mathbf{L}_1 \tag{8.103}$$

En forma sucesiva,

$$\mathbf{A}_3 = \mathbf{L}_2^{-1}\mathbf{A}_2\mathbf{L}_2 = \mathbf{L}_2^{-1}\mathbf{L}_1^{-1}\mathbf{A}_1\mathbf{L}_1\mathbf{L}_2 \tag{8.104}$$

La secuencia de matrices a menudo converge a un bloque de forma triangular superior en el cual a cada bloque le corresponden valores propios de igual módulo. En el caso de una matriz con valores propios reales y distintos, los valores propios aparecen en orden decreciente, bajando por la diagonal de izquierda a derecha cuando la matriz convergió. Así, con tal de que se satisfagan las condiciones para la descomposición triangular, el método anterior da un proceso repetitivo sencillo adecuado para uso computacional. Los vectores propios se obtienen de la matriz original una vez que se conocen los valores propios.

El segundo método incorpora transformaciones de ortogonalidad dentro del procedimiento, ya que esas transformaciones tienen buenas propiedades de estabilidad [Nakamura, 1992], [Maron *et al.*, 1995]. Las matrices se descomponen en un producto $\mathbf{Q}_\mathbf{r}\mathbf{U}_\mathbf{r}$ donde $\mathbf{U}_\mathbf{r}$ es una matriz triangular superior. Así,

www.FreeLibros.me

$$\mathbf{A} = \mathbf{A_1} = \mathbf{Q_1 U_1} \tag{8.105}$$

Generalizando, se tiene que

$$\mathbf{A_r} = \mathbf{U_{r-1} Q_{r-1}} = \mathbf{Q_r U_r}, \qquad r = 2, 3, \ldots \tag{8.106}$$

Se nota que, como antes, las matrices son *similares* debido a que

$$\mathbf{A_r} = \mathbf{U_{r-1} Q_{r-1}} = \mathbf{Q_{r-1}^{-1} A_{r-1} Q_{r-1}} \tag{8.107}$$

Este método es más complicado y toma más tiempo que el método **L**-**R**, pero tiene el beneficio de su gran estabilidad. Como en el método **L**-**R**, las matrices convergen a una forma triangular superior con los valores propios en la diagonal. La descomposición base $\mathbf{A} = \mathbf{Q_r U_r}$ se puede lograr para cualquier matriz, lo que no sucede en el caso de la descomposición **L**-**U**.

8.8 Comparación de métodos

En el caso donde los valores propios son todos distintos, hay un vector propio único que corresponde a cada valor propio. En el caso de un valor propio de multiplicidad *m*, hay tal vez *m* vectores propios o menos. En el último caso habrá menos de *n* vectores propios para la matriz **A**, y los vectores propios no pueden formar una base para el espacio.

Existen dos tipos de métodos que se pueden usar. Los métodos iterativos son fáciles de usar y, en ciertas circunstancias, se puede encontrar una raíz de manera muy efectiva. Estos métodos se pueden modificar para encontrar más de una raíz; pero, en general, no se usarán para encontrar todos los valores propios de una matriz. Los métodos que se usan cuando se necesitan todos los valores propios de una matriz, se basan en transformaciones que reducen la matriz a una forma sencilla que se puede resolver fácilmente para encontrar los valores propios. Si se usan transformaciones de similitud, las nuevas matrices tienen los mismos valores propios que las matrices originales, con una relación sencilla entre los viejos y los nuevos vectores propios.

También se debe tomar en cuenta la estructura de la matriz: si es rala, en bloques o simétrica. Existen métodos de fácil implementación para estos tipos de matrices como, por ejemplo, el método del cociente de Rayleigh o la iteración ortogonal con la aceleración de Ritz, entre otros.

8.9 Programas desarrollados en Matlab

Esta sección proporciona los códigos de los programas desarrollados en Matlab para todos los ejercicios propuestos. A continuación se incluye una lista de todos ellos

8.9.1. Método de Householder
8.9.2. Multiplicación sucesiva por **Y**k
8.9.3. Método de potenciación

8.9.1 Método de Householder

El método reduce una matriz simétrica a una matriz banda (una arriba y otra debajo de la diagonal) mediante transformaciones ortogonales.

Programa principal del método de Householder

```
% Algoritmo de Householder para reducir una matriz simétrica a su forma tridiagonal.
% El programa es general, pero se toma el ejemplo resuelto en el libro para su
% desarrollo.
clear all
```

www.FreeLibros.me

```
clc
% Matriz simétrica A
A = [8     2     5     2     7     2
     2     5     4     7     2     8
     5     4     7     3     5     4
     2     7     3     3     7     9
     7     2     5     7     4     1
     2     8     4     9     1     9 ];
n = rank(A);      % Dimensión de la matriz A.
I = eye(n);       % Matriz identidad del mismo tamaño de A.
% Ciclo iterativo para reducir la matriz a tridiagonal.
for k = 1:n-2
    X           = A(:,k);
    S           = sign(X(k+1))*sqrt(sum(X(k+1:n).^2));
    R           = sqrt(2*S*(X(k+1)+S));
    W(1:k,1)    = zeros;
    W(k+1,1)    = (X(k+1)+S)/R;
    W(k+2:n,1) = X(k+2:n)/R;
    A           = (I-2*W*W.')*A*(I-2*W*W.');
end
% Redondea los resultados con un error de 1e-12.
Af = (round(A*1e12))/1e12;
```

8.9.2 Multiplicación sucesiva por Yk

Con la multiplicación sucesiva de una matriz por un vector **Y**k, se obtiene un vector propio, y el factor de cambio entre dos multiplicaciones se relaciona con el valor propio.

Programa principal de la multiplicación sucesiva por Yk

```
% Cálculo de valores y vectores propios utilizando el método de la multiplicación
% sucesiva por un vector cualesquiera.
clear all
clc
format long g
A  = [7 2 3; 6 5 4; 2 8 9];   % Matriz original.
A0 = A;                       % Asignación de la matriz a otra variable.
yp = [1;7;54];                % Vector inicial para la multiplicación sucesiva.
N  = rank(A);                 % Número de variables del sistema original.
Nt = 1;                       % Variable utilizada para iniciar la convergencia.
To = 1e-12;                   % Tolerancia utilizada para la convergencia.
% Ciclo para el cálculo de los tres valores propios.
for k = 1:N
    % Ciclo de convergencia.
    while Nt > To
        m = max(abs(yp));   % Máximo valor para normalizar el vector.
        yp = yp./m;         % Normalización del vector yp.
        y1 = A*yp;          % Multiplicación sucesiva de la matriz A por yp.
        mt = max(abs(y1));  % Máximo valor de y1.
        Nt = abs(m-mt);     % Cálculo de la diferencia entre dos soluciones
                            % consecutivas.
        yp = y1;            % Asignación del vector nuevo a la variable.
    end
    yp = yp./m;                    % Normalización del vector yp.
    [Ym,Pos] = (max(abs(yp)));     % Se busca el valor máximo de yp y la posición.
    Ac = A - yp*A(Pos,:) ;         % Se calcula la siguiente matriz, la cual tiene un
                                   % valor propio igual a cero.
    A = Ac;                        % Se reasigna la matriz.
    Ep(:,k) = yp;                  % Se guarda el vector propio.
    Epn(:,k) = yp/norm(yp);        % Se normaliza el vector propio.
    Ez(k,1) = m;                   % Se guarda el valor propio.
    XY(k,1) = Pos;                 % Se guarda la posición del valor propio k.
    if k == 1                      % Esta condición es para guardar la segunda matriz,
```

www.FreeLibros.me

```
      A1 = A;                    % se guarda de esta manera.
    end
    Nt = 1;                      % Variable utilizada para iniciar la convergencia.
end
% Cálculo del segundo vector propio.
k1=(Ez(1)-Ez(2))/(A0(XY(1),:)*Ep(:,2));  % Se calcula la constante para referir el
                                          % cálculo a la matriz original.
E2 = Ep(:,1)-k1*Ep(:,2);                 % Se calcula el vector propio referido a la
                                          % matriz original.
E2n = E2/norm(E2);                        % Se normaliza el vector propio.
% Cálculo del tercer vector propio.
k2 = (Ez(2)-Ez(3))/(A(XY(3),:)*Ep(:,3)); % Se calcula la constante para referir el
                                          % cálculo a la matriz anterior.
E3 = Ep(:,2)-k2*Ep(:,3);                 % Se calcula el vector propio referido a la
                                          % matriz anterior.
E3n = E3/norm(E3);                        % Se normaliza el vector propio.
k3 = (Ez(1)-Ez(2))/(A0(XY(2),:)*E3n);    % Se calcula la constante para referir el
                                          % cálculo a la matriz original.
E4 = Ep(:,1)-k3*E3n;                      % Se calcula el vector propio referido a la
                                          % matriz original.
E4n = E4/norm(E4);                        % Se normaliza el vector propio.
% Los valores propios son, por tanto,
Ez
% Los vectores propios son a su vez
Epn(:,1)
E2n
E4n
```

8.9.3 Método de potenciación

El método de potenciación llega a una matriz idempotente que contiene un vector propio mediante la elevación al cuadrado en forma sucesiva de una matriz cualquiera. El factor de cambio entre dos potenciaciones consecutivas se relaciona con el valor propio.

Programa principal del método de potenciación

```
% Este programa calcula el valor propio más grande y su correspondiente vector
% propio. A continuación se hace una deflación para obtener una nueva matriz que
% contiene los dos valores y vectores propios restantes, y así sucesivamente.
%
clc
clear all
format long g
% Matriz de coeficientes.
A = [ 7 2 3; 6 5 4; 2 8 9 ];
% Resultados del método de potenciación.
Nc = length(A);    % Dimensión de la matriz principal.
% Ciclo iterativo para calcular todos los valores y vectores propios.
for k = 1:Nc
   % Función que potencia la matriz hasta obtener el valor propio de mayor módulo
   % y su correspondiente vector propio.
   [Ev,Ld,B] = Df_Potenciacion(A);
   Lr(k)     = Ld;    % Valor propio de mayor módulo.
   Ed(:,k)  = Ev;     % Vector propio asociado al valor propio de mayor módulo.
   A          = B;    % Nueva matriz con los valores propios restantes.
end
```

Función de MatlAb llamada Df_potenciación

```
% Función para calcular el valor propio de mayor módulo y su correspondiente vector
% propio.
function[Ev,Ld,B] = Df_Potenciacion(A);
A0 = A^2;                      % Primera elevación al cuadrado.
```

www.FreeLibros.me

```
A0 = A0./max(max(abs(A0)));     % Normalización.
tol = 1e-12;                    % Tolerancia de convergencia en la potenciación.
mx = 1;                         % Variable que me indica el cumplimiento de la
                                % tolerancia.
% Ciclo while que potencia la matriz hasta cumplir con una tolerancia.
while mx > tol
   A1 = A0^2;                      % Primera potenciación.
   A1 = A1./max(max(abs(A1)));     % Normalización.
   A0 = A1^2;                      % Segunda potenciación.
   A0 = A0./max(max(abs(A0)));     % Normalización.
   mx = max(max(abs(A1-A0)));      % Cálculo del cumplimiento de la tolerancia.
end
% La matriz A0 es proporcional a la matriz idempotente sabiendo que A0 = k*A0^2,
% donde k es el coeficiente de proporcionalidad, por tanto, k = A0./A0^2, y, en
% consecuencia, la matriz idempotente será Id1 = k.*A0.
k    = A0./A0^2;                        % Coeficiente de proporcionalidad.
Idem = k.*A0;                           % Cálculo de la matriz idempotente.
Ld   = A(1,:)*Idem(:,1)/Idem(1,1);      % Valor propio asociado a la idempotente.
% El nuevo sistema de donde se calcula la siguiente matriz idempotente es B = A -
% Ld1*Id1, por tanto:
B = A - Ld*Idem;
% El vector propio es proporcional a los vectores columna de la matriz idempotente.
Ev = Idem(:,1)/norm(Idem(:,1));
```

Problemas propuestos

8.10.1 Calcule los vectores propios y la forma canónica de Jordan de la siguiente matriz

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 1 & 0 & 1 \\ 0 & 1 & 2 & 1 & 1 \\ 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

8.10.2 Calcule los vectores propios y la forma canónica de Jordan de la siguiente matriz

$$\mathbf{A} = \begin{bmatrix} 2 & 4 & 8 & 0 & 3 \\ 0 & 2 & 7 & 1 & 6 \\ 0 & 0 & 2 & 5 & 9 \\ 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

8.10.3 Calcule los vectores propios y la forma canónica de Jordan de la siguiente matriz

$$\mathbf{A} = \begin{bmatrix} 3 & 1 & 1 & 1 & 1 \\ 0 & 3 & 2 & 1 & 2 \\ 0 & 0 & 3 & 2 & 1 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

www.FreeLibros.me

8.10.4 Calcule los vectores propios y la forma canónica de Jordan de la siguiente matriz

$$\mathbf{A} = \begin{bmatrix} 4 & 2 & 3 & 1 & 1 \\ 0 & 4 & 1 & 1 & 2 \\ 0 & 0 & 4 & 1 & 3 \\ 0 & 0 & 0 & 4 & 2 \\ 0 & 0 & 0 & 0 & 4 \end{bmatrix}$$

8.10.5 Calcule los vectores propios y la forma canónica de Jordan de la siguiente matriz

$$\mathbf{A} = \begin{bmatrix} 5 & 1 & 2 & 1 & 0 \\ 0 & 5 & 1 & 0 & 0 \\ 0 & 0 & 5 & 1 & 1 \\ 0 & 0 & 0 & 5 & 2 \\ 0 & 0 & 0 & 0 & 5 \end{bmatrix}$$

8.10.6 Calcule los vectores propios y la forma canónica de Jordan de la siguiente matriz

$$\mathbf{A} = \begin{bmatrix} 6 & 2 & 0 & 0 & 0 \\ 0 & 6 & 1 & 0 & 0 \\ 0 & 0 & 6 & 1 & 1 \\ 0 & 0 & 0 & 6 & 4 \\ 0 & 0 & 0 & 0 & 6 \end{bmatrix}$$

8.10.7 Calcule los vectores propios y la forma canónica de Jordan de la siguiente matriz

$$\mathbf{A} = \begin{bmatrix} 7 & 1 & 1 & 0 & 0 \\ 0 & 7 & 2 & 2 & 0 \\ 0 & 0 & 7 & 3 & 3 \\ 0 & 0 & 0 & 7 & 4 \\ 0 & 0 & 0 & 0 & 7 \end{bmatrix}$$

8.10.8 Calcule los vectores propios y la forma canónica de Jordan de la siguiente matriz

$$\mathbf{A} = \begin{bmatrix} 8 & 2 & 0 & 0 & 0 \\ 0 & 8 & 1 & 0 & 0 \\ 0 & 0 & 8 & 2 & 0 \\ 0 & 0 & 0 & 8 & 1 \\ 0 & 0 & 0 & 0 & 8 \end{bmatrix}$$

8.10.9 Calcule los vectores propios y la forma canónica de Jordan de la siguiente matriz

$$\mathbf{A} = \begin{bmatrix} 9 & 0 & 0 & 1 & 1 \\ 0 & 9 & 0 & 0 & 1 \\ 0 & 0 & 9 & 0 & 0 \\ 0 & 0 & 0 & 9 & 0 \\ 0 & 0 & 0 & 0 & 9 \end{bmatrix}$$

www.FreeLibros.me

8.10.10 Utilizando el método de Householder, reduzca a su forma tridiagonal simétrica la siguiente matriz:

$$\mathbf{A} = \begin{bmatrix} 2 & 4 & 6 & 8 & 1 & 9 \\ 4 & 3 & 9 & 7 & 3 & 1 \\ 6 & 9 & 1 & 2 & 5 & 4 \\ 8 & 7 & 2 & 4 & 9 & 1 \\ 1 & 3 & 5 & 9 & 5 & 8 \\ 9 & 1 & 4 & 1 & 8 & 2 \end{bmatrix}$$

8.10.11 Utilizando el método de Householder, reduzca a su forma tridiagonal simétrica la siguiente matriz:

$$\mathbf{A} = \begin{bmatrix} 5 & 8 & 6 & 4 & 8 & 2 & 3 \\ 8 & 9 & 4 & 9 & 2 & 1 & 8 \\ 6 & 4 & 9 & 7 & 2 & 6 & 8 \\ 4 & 9 & 7 & 8 & 5 & 1 & 9 \\ 8 & 2 & 2 & 5 & 8 & 6 & 9 \\ 2 & 1 & 6 & 1 & 6 & 1 & 6 \\ 3 & 8 & 8 & 9 & 9 & 6 & 7 \end{bmatrix}$$

8.10.12 Utilizando el método de Householder, reduzca a su forma tridiagonal simétrica la siguiente matriz:

$$\mathbf{A} = \begin{bmatrix} 2 & 9 & 3 & 7 & 1 & 6 & 4 & 8 \\ 9 & 5 & 3 & 9 & 4 & 2 & 5 & 6 \\ 3 & 3 & 5 & 9 & 4 & 5 & 6 & 7 \\ 7 & 9 & 9 & 3 & 5 & 9 & 6 & 4 \\ 1 & 4 & 4 & 5 & 7 & 6 & 2 & 9 \\ 6 & 2 & 5 & 9 & 6 & 3 & 8 & 4 \\ 4 & 5 & 6 & 6 & 2 & 8 & 7 & 1 \\ 8 & 6 & 7 & 4 & 9 & 4 & 1 & 5 \end{bmatrix}$$

8.10.13 Utilizando el método de Householder, reduzca a su forma tridiagonal simétrica la siguiente matriz:

$$\mathbf{A} = \begin{bmatrix} 5 & 2 & 6 & 8 & 9 & 1 & 4 & 6 & 8 \\ 2 & 4 & 7 & 2 & 7 & 2 & 1 & 8 & 3 \\ 6 & 7 & 2 & 1 & 5 & 8 & 2 & 7 & 5 \\ 8 & 2 & 1 & 1 & 6 & 5 & 2 & 9 & 1 \\ 9 & 7 & 5 & 6 & 9 & 1 & 1 & 4 & 5 \\ 1 & 2 & 8 & 5 & 1 & 4 & 2 & 5 & 9 \\ 4 & 1 & 2 & 2 & 1 & 2 & 6 & 1 & 4 \\ 6 & 8 & 7 & 9 & 4 & 5 & 1 & 9 & 2 \\ 8 & 3 & 5 & 1 & 5 & 9 & 4 & 2 & 8 \end{bmatrix}$$

www.FreeLibros.me

8.10.14 Utilizando el método de Householder, reduzca a su forma tridiagonal simétrica la siguiente matriz:

$$\mathbf{A}=\begin{bmatrix}2&5&6&8&1&9&5&4&7&6\\5&2&8&3&7&9&1&6&5&4\\6&8&1&7&6&4&9&3&2&8\\8&3&7&6&2&9&4&2&8&6\\1&7&6&2&9&6&1&3&8&4\\9&9&4&9&6&7&4&6&3&2\\5&1&9&4&1&4&6&8&7&2\\4&6&3&2&3&6&8&4&5&2\\7&5&2&8&8&3&7&5&1&6\\6&4&8&6&4&2&2&2&6&7\end{bmatrix}$$

8.10.15 Utilizando el método de Householder, reduzca a su forma tridiagonal simétrica la siguiente matriz:

$$\mathbf{A}=\begin{bmatrix}2&5&7&6&1&3&5&4&6&9&5\\5&1&8&4&9&7&6&2&3&4&8\\7&8&4&8&6&3&2&1&7&8&4\\6&4&8&1&3&8&6&5&4&7&7\\1&9&6&3&7&5&1&2&8&4&6\\3&7&3&8&5&2&9&7&6&2&5\\5&6&2&6&1&9&4&6&2&8&7\\4&2&1&5&2&7&6&2&8&4&3\\6&3&7&4&8&6&2&8&1&6&8\\9&4&8&7&4&2&8&4&6&7&1\\5&8&4&7&6&5&7&3&8&1&4\end{bmatrix}$$

8.10.16 Utilizando el método de Householder, reduzca a su forma tridiagonal simétrica la siguiente matriz:

$$\mathbf{A}=\begin{bmatrix}5&8&3&9&4&6&2&8&9&4&1&6&7\\8&2&4&3&9&8&4&5&3&2&2&8&7\\3&4&1&5&9&6&3&8&6&5&2&6&5\\9&3&5&1&2&5&8&9&4&6&7&1&3\\4&9&9&2&3&5&6&9&8&4&1&5&7\\6&8&6&5&5&2&4&1&8&6&5&9&7\\2&4&3&8&6&4&1&5&4&9&7&5&2\\8&5&8&9&9&1&5&2&7&6&5&3&1\\9&3&6&4&8&8&4&7&1&4&6&8&2\\4&2&5&6&4&6&9&6&4&3&7&5&4\\1&2&2&7&1&5&7&5&6&7&2&8&6\\6&8&6&1&5&9&5&3&8&5&8&1&2\\7&7&5&3&7&7&2&1&2&4&6&2&8\end{bmatrix}$$

www.FreeLibros.me

8.10.17 Utilizando el método de Householder, reduzca a su forma tridiagonal simétrica la siguiente matriz:

$$\mathbf{A}=\begin{bmatrix}1&3&5&7&9&2&4&6&8&2&5&9&4&5\\3&2&5&8&6&4&9&4&7&1&6&2&5&7\\5&5&2&8&6&5&1&4&7&6&5&9&2&5\\7&8&8&1&2&5&8&9&4&7&6&3&2&9\\9&6&6&2&1&8&6&5&7&3&1&7&5&2\\2&4&5&5&8&2&7&6&3&9&1&8&2&4\\4&9&1&8&6&7&1&9&6&4&7&2&8&1\\6&4&4&9&5&6&9&2&8&4&3&8&1&6\\8&7&7&4&7&3&6&8&2&5&6&8&1&7\\2&1&6&7&3&9&4&4&5&1&5&7&9&2\\5&6&5&6&1&1&7&3&6&5&1&9&7&2\\9&2&9&3&7&8&2&8&8&7&9&3&6&1\\4&5&2&2&5&2&8&1&1&9&7&6&4&7\\5&7&5&9&2&4&1&6&7&2&2&1&7&3\end{bmatrix}$$

8.10.18 Utilizando el método de Householder, reduzca a su forma tridiagonal simétrica la siguiente matriz:

$$\mathbf{A}=\begin{bmatrix}5&8&9&3&2&7&5&6&9&4&1&8&5&2&4\\8&5&2&8&6&4&9&7&1&5&6&3&5&4&6\\9&2&2&5&4&8&6&2&5&8&4&1&5&6&9\\3&8&5&2&9&6&5&8&4&1&7&5&6&3&2\\2&6&4&9&2&5&6&3&9&7&5&2&4&6&7\\7&4&8&6&5&1&6&9&5&3&8&7&5&2&6\\5&9&6&5&6&6&4&7&3&6&2&8&9&7&1\\6&7&2&8&3&9&7&1&6&2&8&9&3&4&6\\9&1&5&4&9&5&3&6&1&4&8&3&6&7&4\\4&5&8&1&7&3&6&2&4&6&9&5&1&2&7\\1&6&4&7&5&8&2&8&8&9&2&6&4&9&4\\8&3&1&5&2&7&8&9&3&5&6&2&5&7&3\\5&5&5&6&4&5&9&3&6&1&4&5&2&6&9\\2&4&6&3&6&2&7&4&7&2&9&7&6&1&4\\4&6&9&2&7&6&1&6&4&7&4&3&9&4&1\end{bmatrix}$$

8.10.19 Utilizando el método de multiplicación sucesiva por un vector cualquiera $\mathbf{Y}=\begin{bmatrix}1 & 2 & 3\end{bmatrix}^T$, calcule el valor propio de mayor magnitud y su correspondiente vector propio, si se tiene la siguiente matriz:

$$\mathbf{A}=\begin{bmatrix}7&2&1\\1&9&2\\2&3&7\end{bmatrix}$$

www.FreeLibros.me

8.10.20 Utilizando el método de multiplicación sucesiva por un vector cualquiera $\mathbf{Y} = [1 \quad 3 \quad 6]^T$, calcule el valor propio de mayor magnitud y su correspondiente vector propio, si se tiene la siguiente matriz:

$$\mathbf{A} = \begin{bmatrix} 3 & 1 & 7 \\ 2 & 6 & 1 \\ 5 & 2 & 6 \end{bmatrix}$$

8.10.21 Utilizando el método de multiplicación sucesiva por un vector cualquiera $\mathbf{Y} = [1 \quad 3 \quad 6]^T$, calcule el valor propio de mayor magnitud y su correspondiente vector propio, si se tiene la siguiente matriz:

$$\mathbf{A} = \begin{bmatrix} 7 & 1 & 4 \\ 2 & 7 & 1 \\ 3 & 2 & 8 \end{bmatrix}$$

8.10.22 Utilizando el método de multiplicación sucesiva por un vector cualquiera $\mathbf{Y} = [1 \quad 3 \quad 6]^T$, calcule el valor propio de mayor magnitud y su correspondiente vector propio, si se tiene la siguiente matriz:

$$\mathbf{A} = \begin{bmatrix} 8 & 1 & 1 \\ 2 & 7 & 2 \\ 3 & 3 & 9 \end{bmatrix}$$

8.10.23 Utilizando el método de multiplicación sucesiva por un vector cualquiera $\mathbf{Y} = [1 \quad 3 \quad 6]^T$, calcule el valor propio de mayor magnitud y su correspondiente vector propio, si se tiene la siguiente matriz:

$$\mathbf{A} = \begin{bmatrix} 6 & 1 & 2 \\ 3 & 7 & 4 \\ 5 & 6 & 8 \end{bmatrix}$$

8.10.24 Utilizando el método de multiplicación sucesiva por un vector cualquiera $\mathbf{Y} = [1 \quad 2 \quad 3]^T$, calcule el valor propio de mayor magnitud y su correspondiente vector propio, si se tiene la siguiente matriz:

$$\mathbf{A} = \begin{bmatrix} 9 & 1 & 2 \\ 1 & 6 & 2 \\ 1 & 2 & 7 \end{bmatrix}$$

8.10.25 Utilizando el método de multiplicación sucesiva por un vector cualquiera $\mathbf{Y} = [1 \quad 2 \quad 3 \quad 6]^T$, calcule el valor propio de mayor magnitud y su correspondiente vector propio, si se tiene la siguiente matriz:

$$\mathbf{A} = \begin{bmatrix} 3 & 6 & 1 & 7 \\ 2 & 6 & 8 & 1 \\ 4 & 6 & 9 & 2 \\ 5 & 7 & 2 & 6 \end{bmatrix}$$

www.FreeLibros.me

8.10.26 Utilizando el método de multiplicación sucesiva por un vector cualquiera $\mathbf{Y} = [1 \quad 2 \quad 3 \quad 6]^T$, calcule el valor propio de mayor magnitud y su correspondiente vector propio, si se tiene la siguiente matriz:

$$\mathbf{A} = \begin{bmatrix} 9 & 2 & 1 & 6 \\ 2 & 8 & 1 & 2 \\ 3 & 2 & 9 & 1 \\ 4 & 1 & 2 & 7 \end{bmatrix}$$

8.10.27 Utilizando el método de multiplicación sucesiva por un vector cualquiera $\mathbf{Y} = [1 \quad 2 \quad 3 \quad 6]^T$, calcule el valor propio de mayor magnitud y su correspondiente vector propio, si se tiene la siguiente matriz:

$$\mathbf{A} = \begin{bmatrix} 7 & 1 & 2 & 3 \\ 1 & 8 & 2 & 3 \\ 2 & 2 & 9 & 3 \\ 1 & 2 & 3 & 6 \end{bmatrix}$$

8.10.28 Utilizando el método de potenciación, determine los valores propios y sus vectores propios asociados, si se tiene la siguiente matriz. Haga la comprobación utilizando la fórmula (8.93).

$$\mathbf{A} = \begin{bmatrix} 9 & 1 & 1 \\ 1 & 8 & 2 \\ 2 & 3 & 7 \end{bmatrix}$$

8.10.29 Utilizando el método de potenciación, determine los valores propios y sus vectores propios asociados, si se tiene la siguiente matriz. Haga la comprobación utilizando la fórmula (8.93).

$$\mathbf{A} = \begin{bmatrix} 9 & 5 & 1 \\ 2 & 8 & 3 \\ 2 & 4 & 7 \end{bmatrix}$$

8.10.30 Utilizando el método de potenciación, determine los valores propios y sus vectores propios asociados, si se tiene la siguiente matriz. Haga la comprobación utilizando la fórmula (8.93).

$$\mathbf{A} = \begin{bmatrix} 6 & 1 & 1 \\ 2 & 5 & 2 \\ 1 & 2 & 7 \end{bmatrix}$$

8.10.31 Utilizando el método de potenciación, determine los valores propios y sus vectores propios asociados, si se tiene la siguiente matriz. Haga la comprobación utilizando la fórmula (8.93).

$$\mathbf{A} = \begin{bmatrix} 7 & 1 & 2 & 1 \\ 2 & 6 & 1 & 1 \\ 3 & 1 & 7 & 1 \\ 1 & 2 & 3 & 8 \end{bmatrix}$$

www.FreeLibros.me

8.10.32 Utilizando el método de potenciación, determine los valores propios y sus vectores propios asociados, si se tiene la siguiente matriz. Haga la comprobación utilizando la fórmula (8.93).

$$\mathbf{A}=\begin{bmatrix}9&2&2&1\\1&6&3&1\\1&2&8&1\\2&3&1&7\end{bmatrix}$$

8.10.33 Utilizando el método de potenciación, determine los valores propios y sus vectores propios asociados, si se tiene la siguiente matriz. Haga la comprobación utilizando la fórmula (8.93).

$$\mathbf{A}=\begin{bmatrix}9&1&2&1\\2&7&3&2\\1&1&8&2\\1&2&1&9\end{bmatrix}$$

8.10.34 Utilizando el método de potenciación, determine los valores propios y sus vectores propios asociados, si se tiene la siguiente matriz. Haga la comprobación utilizando la fórmula (8.93).

$$\mathbf{A}=\begin{bmatrix}8&1&2&1\\2&7&1&3\\3&1&9&4\\2&1&2&7\end{bmatrix}$$

8.10.35 Utilizando el método de potenciación, determine los valores propios y sus vectores propios asociados, si se tiene la siguiente matriz. Haga la comprobación utilizando la fórmula (8.93).

$$\mathbf{A}=\begin{bmatrix}9&1&1&1\\1&7&2&1\\2&1&7&2\\2&1&2&8\end{bmatrix}$$

8.10.36 Utilizando el método de potenciación, determine los valores propios y sus vectores propios asociados, si se tiene la siguiente matriz. Haga la comprobación utilizando la fórmula (8.93).

$$\mathbf{A}=\begin{bmatrix}5&2&6&4\\1&5&9&7\\3&5&4&7\\5&6&9&4\end{bmatrix}$$

www.FreeLibros.me

Capítulo 9

Ecuaciones diferenciales parciales

9.1 Introducción

El tratamiento numérico de las ecuaciones diferenciales parciales (EDP) es, por sí mismo, un tema amplio. Las EDP están presentes en la mayoría de las representaciones matemáticas que modelan la evolución temporal y espacial de los sistemas físicos continuos, tales como la mecánica de fluidos y los campos electromagnéticos, entre otros.

Las EDP están clasificadas en tres categorías con base en sus características o curvas de propagación. Éstas son:

1. Hiperbólicas.
2. Parabólicas.
3. Elípticas.

A continuación se da un breve ejemplo de cada una de ellas.

9.1.1 Hiperbólicas

Un ejemplo prototipo de una ecuación hiperbólica es la ecuación de onda de una sola dimensión, la cual se representa con la siguiente ecuación:

$$\frac{\partial^2 u}{\partial t^2} = v^2 \frac{\partial^2 u}{\partial x^2} \tag{9.1}$$

donde v es la velocidad de propagación de la onda y se mantiene constante.

9.1.2 Parabólicas

La EDP representativa de las ecuaciones parabólicas es la ecuación de difusión,

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(D \frac{\partial u}{\partial x} \right) \tag{9.2}$$

donde $D > 0$ es el coeficiente de difusión.

www.FreeLibros.me

9.1.3 Elípticas

La ecuación elíptica prototipo es la *ecuación de Poisson*,

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=\rho(x,y) \tag{9.3}$$

donde ρ es la fuente conocida; si $\rho(x,y)=0$, la ecuación se conoce como *ecuación de Laplace*.

Desde el punto de vista computacional, la clasificación canónica en estos tres tipos no es significativa, o al menos no tan importante como algunas otras condiciones, tales como los problemas de valor inicial o de Cauchy definidos por las EDP hiperbólicas y parabólicas, o los problemas de valor de frontera (o limítrofe) denotados por las EDP elípticas.

En problemas de valor inicial, la información de la variable dependiente u en esquemas de primer orden, y adicionalmente su derivada respecto al tiempo para esquemas de segundo orden, están dadas en un tiempo inicial t_0 para toda x y, entonces, las ecuaciones describen cómo se propaga $u(x,t)$ a través del tiempo. Así, el objetivo de la implementación numérica es calcular esa evolución con alguna precisión adecuada.

En cambio, en problemas de valor de frontera, se encuentra una sola función estática $u(x,y)$ que satisface la ecuación dentro de una región de interés (x,y). Por consiguiente, la meta de la implementación numérica es buscar la convergencia a la solución correcta en todas las partes de manera simultánea. La figura 9.1 enfatiza esta distinción.

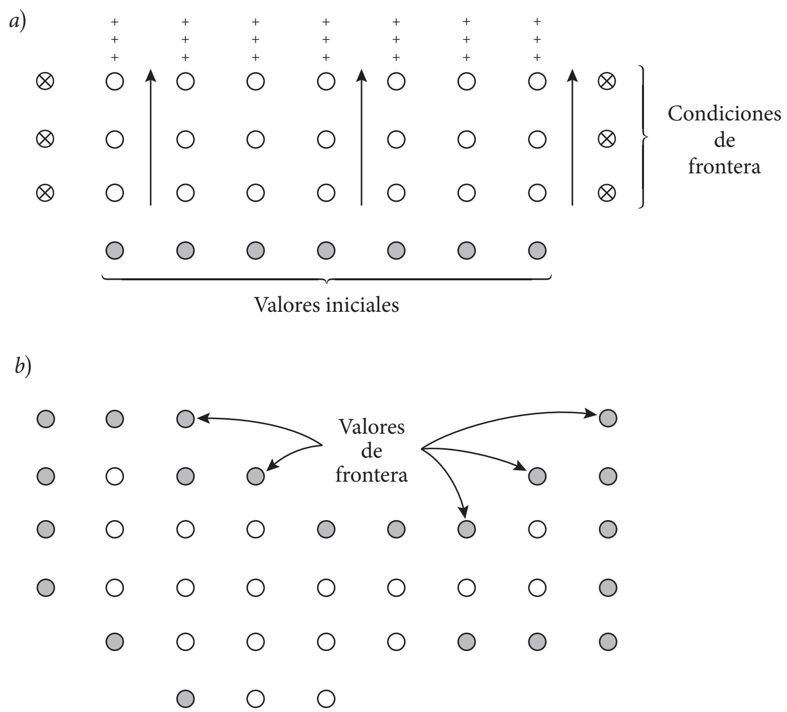


Figura 9.1 Contraste entre un problema de valor inicial *a*) y el problema de valor de la frontera *b*). En *a*) los valores iniciales están dados como una "rebanada de tiempo", apropiada para buscar la solución conforme transcurre éste. En *b*), los valores de la frontera se especifican alrededor de los bordes de una malla, y se utiliza un proceso iterativo para encontrar los valores de todos los puntos internos (círculos abiertos).

Se hace notar que, desde el punto de vista computacional, la subclasificación de problemas de valor inicial en parabólicos e hiperbólicos es poco útil debido a que

a) muchos problemas reales son de tipo mixto, y
b) la mayoría de los problemas hiperbólicos mezclan secciones parabólicas en ellos.

www.FreeLibros.me

Por tanto, resulta más práctico discutir esquemas computacionales que resuelven estos problemas simultáneamente.

Los problemas de valor inicial responden a las siguientes preguntas:

1. ¿Cuáles son las variables dependientes que se propagan con el tiempo?
2. ¿Cuál es la ecuación de evolución para cada variable?
3. ¿Cuál es la derivada más alta, con respecto del tiempo, contenida en la ecuación de evolución de cada variable?
4. ¿Cuáles son las ecuaciones especiales, es decir, las condiciones de frontera que gobiernan la evolución en el tiempo de los puntos de frontera de la región espacial de interés?

Por otro lado, los problemas de valor de frontera definen lo siguiente:

1. ¿Cuáles son las variables?
2. ¿Cuáles ecuaciones se satisfacen en el interior de la región de interés?
3. ¿Cuáles ecuaciones se satisfacen (con puntos sobre) en el confín de la región de interés?

En referencia a los algoritmos para resolver EDP, se deben considerar tres aspectos fundamentales: consistencia, convergencia y estabilidad. La consistencia se refiere a la similitud entre el esquema numérico y la ecuación diferencial que se está tratando de resolver. La convergencia asegura que al disminuir el tamaño de paso considerado, la solución del esquema numérico convergerá a la solución de la ecuación diferencial. Por último, la estabilidad nos asegura que pequeños cambios en las condiciones iniciales implican pequeños cambios en la solución numérica dada por el esquema numérico. Un último punto que se debe considerar en los esquemas propuestos es la facilidad de implementación computacional y el costo en la ejecución del esquema.

Las siguientes secciones tratan de la estabilidad de las implementaciones numéricas desde el punto de vista computacional. En el caso de los problemas de valor de frontera, la estabilidad es relativamente fácil de lograr; por tanto, la preocupación principal es la eficiencia de los algoritmos y los requisitos de almacenamiento, debido a que todas las condiciones de un problema se deben satisfacer simultáneamente. Como problema modelo se considera la solución de la ecuación (9.3) por el método de diferencias finitas. Se representa la función $u(x, y)$ por sus valores en el conjunto de puntos siguiente:

$$\begin{aligned} x_j &= x_0 + j\Delta x, \qquad j = 0, 1, \ldots, J \\ y_l &= y_0 + l\Delta y, \qquad l = 0, 1, \ldots, L \end{aligned} \tag{9.4}$$

donde Δx y Δy son los espaciamientos de la malla en los ejes respectivos.

Si se denota $u_{j,l}$ a la aproximación de $u(x_j, y_l)$ y $\rho_{j,l}$ a $\rho(x_j, y_l)$ y, si se considera que el espaciamiento de la malla en ambas direcciones es idéntico, esto es, si $\Delta x = \Delta y = \Delta$, la representación en diferencias finitas de la ecuación (9.3) es

$$\frac{u_{j+1,l} - 2u_{j,l} + u_{j-1,l}}{\Delta^2} + \frac{u_{j,l+1} - 2u_{j,l} + u_{j,l-1}}{\Delta^2} = \rho_{j,l} \tag{9.5}$$

o, de manera equivalente,

$$u_{j+1,l} + u_{j-1,l} + u_{j,l+1} + u_{j,l-1} - 4u_{j,l} = \Delta^2 \rho_{j,l} \tag{9.6}$$

Esto se ilustra en forma gráfica en la figura 9.2. Enumerando las dos dimensiones de los puntos en la malla, en una secuencia unidimensional; se puede escribir este sistema de ecuaciones lineales en forma matricial, es decir, si se tiene que

$$i \equiv j(L+1)+1 \qquad \text{para} \qquad j = 1, \ldots, J \qquad \text{y} \qquad l = 1, \ldots, L \tag{9.7}$$

www.FreeLibros.me

En otras palabras, i se incrementa más rápido a lo largo de las columnas, representando los valores de y.

Así, la ecuación (9.6) ahora es

$$u_{i+L+1} + u_{i-(L+1)} + u_{i+1} + u_{i-1} - 4u_i = \Delta^2 \rho_i \tag{9.8}$$

Esta ecuación se cumple sólo para los puntos interiores $j = 1, \ldots, J-1$ y $l = 1, \ldots, L-1$. Los puntos de la frontera donde se especifica u o su derivada son:

$$\begin{aligned} j &= 0 && [\text{i.e., } i = 0, \ldots, L] \\ j &= J && [\text{i.e., } i = J(L+1), \ldots, J(L+1)+L] \\ l &= 0 && [\text{i.e., } i = 0, L+1, \ldots, J(L+1)] \\ l &= L && [\text{i.e., } i = L, L+1+L, \ldots, J(L+1)+L] \end{aligned} \tag{9.9}$$

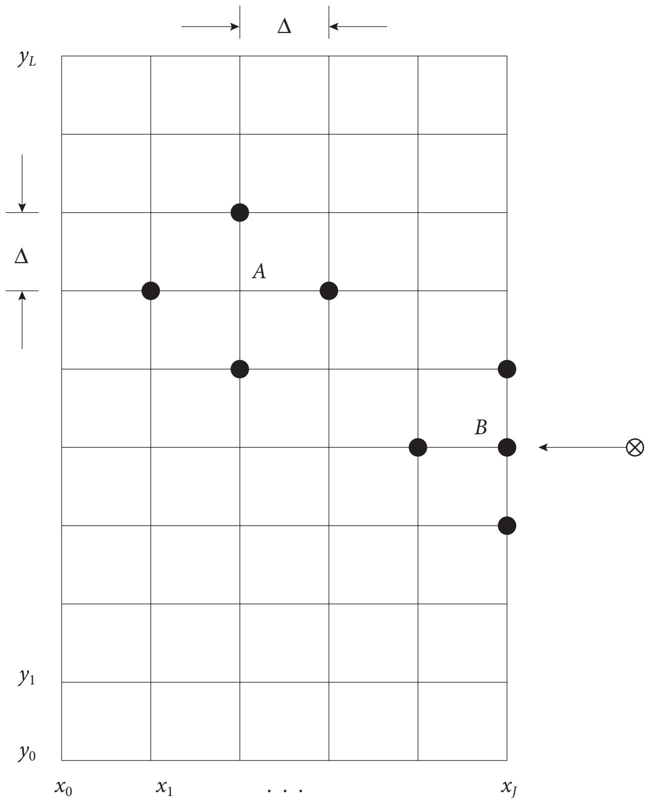


Figura 9.2 Representación en diferencias finitas de una ecuación elíptica de segundo orden sobre una malla bidimensional. Las segundas derivadas en el punto A se evalúan utilizando los valores vecinos a este punto. Las segundas derivadas en el punto B se evalúan utilizando a sus vecinos y a los puntos en la frontera marcados por ⊗.

Si los datos conocidos se pasan del lado derecho de la ecuación (9.8) y se agrupan vectorialmente, entonces la ecuación toma la forma

$$\mathbf{Au} = \mathbf{b} \tag{9.10}$$

En la ecuación (9.10), **A** se llama *matriz tridiagonal a bloques*. Ésta tiene la estructura que se muestra en la figura 9.3. Para el caso de una ecuación elíptica lineal de segundo orden, ésta conduce a una matriz similar (con la misma estructura), excepto que las entradas no nulas son constantes; la ecuación tiene la forma

www.FreeLibros.me

$$a(x,y)\frac{\partial^2 u}{\partial x^2}+b(x,y)\frac{\partial u}{\partial x}+c(x,y)\frac{\partial^2 u}{\partial y^2}+d(x,y)\frac{\partial u}{\partial y}+e(x,y)\frac{\partial^2 u}{\partial x \partial y}+f(x,y)u=g(x,y) \quad \textbf{(9.11)}$$

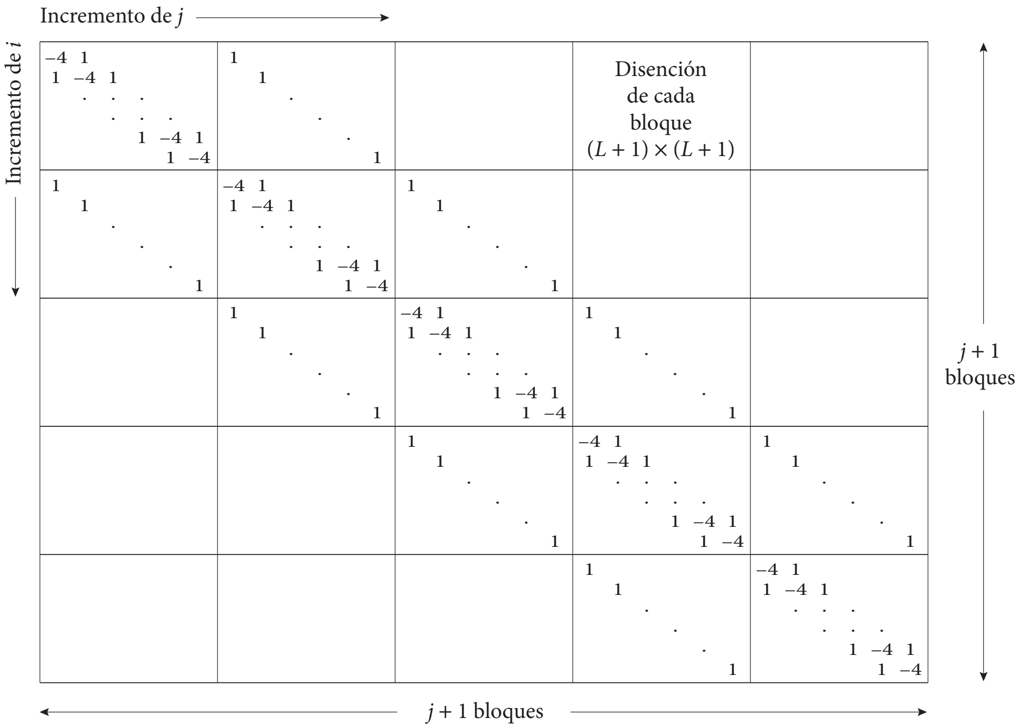


Figura 9.3 Estructura de una matriz que se obtiene de una ecuación elíptica de segundo orden. No todos los elementos son cero. La matriz tiene bloques diagonales que son por sí mismos tridiagonales. Ésta se llama *tridiagonal de bloques*.

9.1.4 Métodos de solución de la ecuación Au = b

Como clasificación preliminar, existen tres métodos para aproximar la solución de la ecuación (9.10):

1. Métodos de relajación.
2. Métodos rápidos como los de Fourier.
3. Métodos directos matriciales.

9.1.4.1 Métodos de relajación

Los métodos de relajación hacen uso de la estructura de una matriz dispersa **A**. El método separa la matriz **A** en dos partes, de la siguiente forma

$$\mathbf{A}=\mathbf{E}-\mathbf{F} \quad \textbf{(9.12)}$$

donde **E** se elige de tal manera que sea fácilmente invertible y **F** se determina a partir de **E**; así, sustituyendo (9.12) en (9.10) se obtiene

$$\mathbf{Eu}=\mathbf{Fu}+\mathbf{b} \quad \textbf{(9.13)}$$

El método de relajación se basa en proponer un vector inicial $\mathbf{u}^{(0)}$ y resolver la ecuación (9.13) iterativamente de tal forma que se obtiene la ecuación

www.FreeLibros.me

$$\mathbf{E}\mathbf{u}^{(r)} = \mathbf{F}\mathbf{u}^{(r-1)} + \mathbf{b} \tag{9.14a}$$

La ecuación para obtener $\mathbf{u}^{(r)}$ es, entonces,

$$\mathbf{u}^{(r)} = \mathbf{E}^{-1}\left(\mathbf{F}\mathbf{u}^{(r-1)} + \mathbf{b}\right) \tag{9.14b}$$

Como **E** es fácilmente invertible, la aplicación del método es sencilla y rápida. Una ventaja de este tipo de métodos es que se detienen cuando se alcanza la precisión definida, en este caso dada por la ecuación (9.8).

9.1.4.2 Métodos rápidos

Los métodos rápidos se aplican sólo a una clase especial de ecuaciones: a las que tienen coeficientes constantes, o, en forma más general, a las que son separables. Además, los confines deben coincidir con los ejes coordenados. Los métodos de esta clase se abordan en la sección 9.4.

9.1.4.3 Métodos directos matriciales

Los métodos matriciales resuelven el sistema

$$\mathbf{A}\mathbf{u} = \mathbf{b} \tag{9.15}$$

Directamente, entre otros, se pueden usar el método de inversa, el de factorización o el de eliminación gaussiana. El grado para el cual uno u otro método resulta práctico depende de la estructura exacta de la matriz **A**, es decir del grado de dispersión que tenga, de su simetría, y de si ésta es definida positiva. Por supuesto, si se tiene más capacidad de almacenamiento y de procesamiento que la que se necesita para el sistema, se usa directamente la inversión. Siempre existirá la disyuntiva de cuál es un tiempo prohibitivo y cuál no. Esta decisión dependerá del uso y de las necesidades específicas; es decir, si un programa se ejecuta en una hora, para un académico puede ser un tiempo razonable y para un empresario inaceptable. Por esta razón, definir lo adecuado será totalmente heurístico.

9.2 Problemas de valor inicial

En el espacio unidimensional, hay una gran clase de EDP que se pueden agrupar como ecuaciones de flujo conservativo. Éstas tienen la estructura:

$$\frac{\partial \mathbf{u}}{\partial t} = -\frac{\partial \mathbf{F}(\mathbf{u})}{\partial x}, \tag{9.16}$$

donde **u** es el vector de incógnitas y **F** es el llamado vector conservador del flujo; **F** puede depender tanto de **u** como de sus derivadas espaciales.

Un ejemplo, prototipo de la ecuación de flujo conservativo (9.16), es el siguiente:

$$\frac{\partial u}{\partial t} = -v\frac{\partial u}{\partial x} \tag{9.17}$$

donde v es constante. De antemano se sabe que la solución analítica es una onda que se propaga en la dirección positiva de x. Ésta se expresa como

$$u = f(x - vt), \tag{9.18}$$

donde f es una función arbitraria.

La estrategia numérica más directa para aproximar la ecuación (9.17) es seleccionar puntos igualmente espaciados a lo largo de los ejes x y t. Así se denota

www.FreeLibros.me

$$x_j = x_0 + j\Delta x, \qquad j = 0, 1, \ldots, J \tag{9.19a}$$

$$t_n = t_0 + n\Delta t, \qquad n = 0, 1, \ldots, N \tag{9.19b}$$

Si se utiliza el método de Euler hacia delante para representar la derivada respecto al tiempo, se obtiene

$$\left.\frac{\partial u}{\partial t}\right|_{j,n} = \frac{u_j^{n+1} - u_j^n}{\Delta t} + O(\Delta t), \tag{9.20}$$

donde la notación u_j^n corresponde a $u(t_n, x_j)$.

Si para la derivada espacial se utiliza la representación de diferencias centrales, se llega a

$$\left.\frac{\partial u}{\partial x}\right|_{j,n} = \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} + O(\Delta x^2) \tag{9.21}$$

Sustituyendo las ecuaciones (9.20) y (9.21) en la ecuación (9.17), se llega a una aproximación de diferencias finitas conocida como *FTCS* (del inglés *Forward Time Centered Space* [espacio centrado en el tiempo de avance]); así se obtiene

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -v\left(\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta t}\right) \tag{9.22}$$

El esquema FTCS se ilustra en la figura 9.4. Éste es un ejemplo elegante de un algoritmo que es fácil de derivar, que requiere un almacenamiento mínimo y permite una ejecución rápida. La representación FTCS es un esquema explícito. Esto indica que u_j^{n+1} se puede calcular para cada *j* a partir de las cantidades conocidas que están a su alrededor. Este algoritmo también es un ejemplo de un esquema de nivel sencillo, ya que sólo se necesitan los valores al nivel *n* para encontrar los valores al nivel $n+1$.

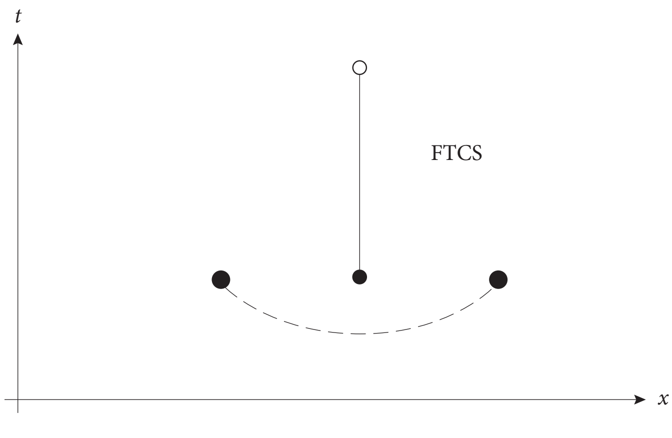


Figura 9.4 Representación del esquema FTCS.

9.2.1 Análisis de estabilidad de Von Neumann

El *análisis de Von Neumann* es local; es decir, si los coeficientes de las ecuaciones en diferencias divergen tan lentamente que se pueden considerar constantes en espacio y tiempo, en este caso las soluciones independientes (valores propios) de las ecuaciones en diferencias son todas de la forma

$$u_j^n = \xi^n e^{ikj\Delta x}, \tag{9.23}$$

www.FreeLibros.me

donde k es un número real de la onda espacial y $\xi = \xi(k)$ es un número complejo que depende de k. Si $|\xi(k)| > 1$ para algún k, entonces las ecuaciones en diferencias son inestables; así, el número ξ se llama *factor de amplificación* para algún número k de onda dada.

Para encontrar $\xi(k)$, se sustituye la ecuación (9.23) en la ecuación (9.22), la cual, dividiendo entre ξ^n, lleva a

$$\xi(k) = 1 - i\frac{v\Delta t}{\Delta x}\operatorname{sen} k\Delta x \tag{9.24}$$

cuyo módulo es mayor que uno para todo k, por lo que el esquema FTCS es incondicionalmente inestable. A pesar de la falta de rigor, el método de Von Neumann por lo general da respuestas válidas y es más fácil de aplicar que otros métodos más minuciosos.

9.2.2 Método de Lax

La inestabilidad en el esquema FTCS se puede eliminar mediante un cambio simple, el cual consiste en reemplazar el término u_j^n en el término de la derivada temporal por su promedio en el espacio (véase la figura 9.5). Así se obtiene

$$u_j^n \to \frac{1}{2}\left(u_{j+1}^n + u_{j-1}^n\right) \tag{9.25}$$

Esto transforma la ecuación (9.22) en

$$u_j^{n+1} = \frac{1}{2}\left(u_{j+1}^n + u_{j-1}^n\right) - \frac{v\Delta t}{2\Delta x}\left(u_{j+1}^n - u_{j-1}^n\right) \tag{9.26}$$

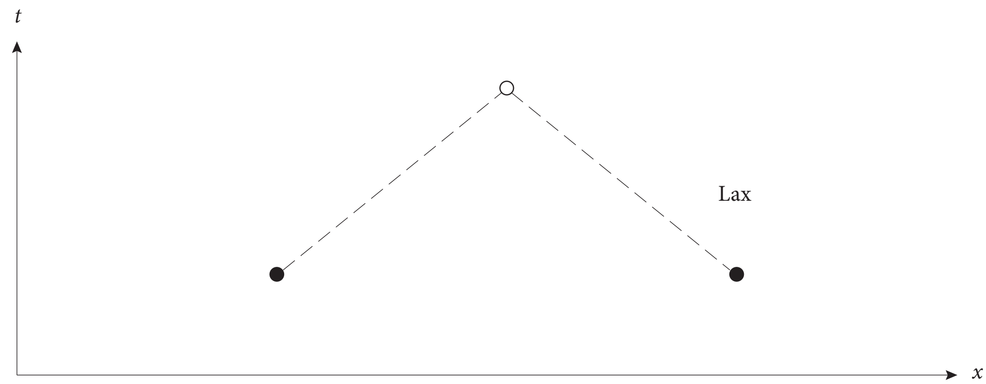


Figura 9.5 Representación del esquema diferenciador Lax. El criterio de estabilidad para este esquema es la *condición de Courant*.

Sustituyendo la ecuación (9.26) en la (9.23), se encuentra el factor de amplificación

$$\xi = \cos k\Delta x - i\frac{v\Delta t}{\Delta x}\operatorname{sen} k\Delta x \tag{9.27}$$

La condición de estabilidad $|\xi| \le 1$ lleva al requisito de que

$$\frac{|v|\Delta t}{\Delta x} \le 1 \tag{9.28}$$

La ecuación (9.28) se llama *criterio de estabilidad de Courant-Friedrichs-Lewy*. Intuitivamente, la condición de estabilidad se puede apreciar en forma gráfica, como lo presenta la figura 9.6. Aquí se muestra cómo la cantidad u_j^{n+1}, de la ecuación (9.26), se calcula a partir de la información de los puntos $j-1$ y

www.FreeLibros.me

$j+1$ en el tiempo n; es decir, x_{j-1} y x_{j+1} son los bordes de la región espacial que permite comunicar la información a u_j^{n+1}. Si el punto u_j^{n+1} se encuentra en el exterior de la región sombreada de la figura 9.6, entonces se requiere información de puntos más distantes de lo que permite el esquema diferenciador. La falta de esta información da pie a la inestabilidad. Por tanto, Δt no puede tomar cualquier valor.

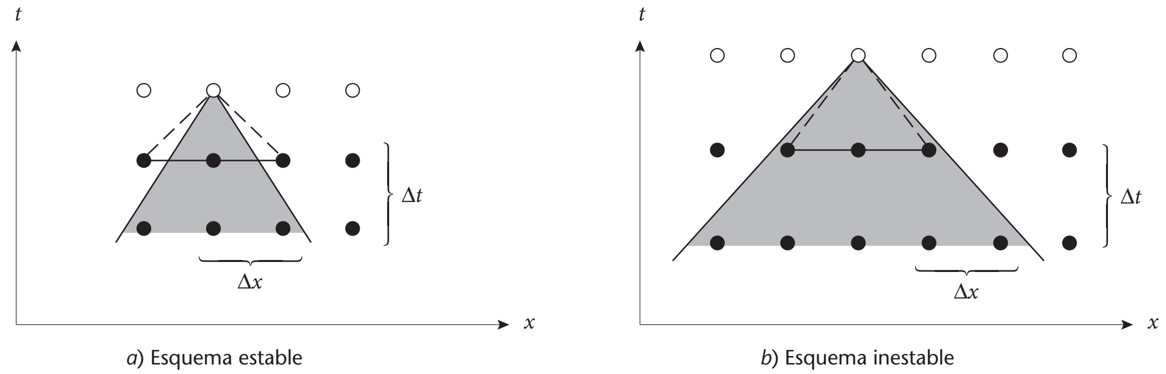


Figura 9.6 Condición de Courant para un esquema de diferenciación. Un esquema diferenciador de Courant es estable si el dominio de dependencia es más grande que el de la EDP, como en *a*); y es inestable si la relación es inversa, como *b*).

9.2.2.1 Criterio de estabilidad de Von Neumann para el método de Lax

Cuando la variable independiente **u** es un vector, el análisis de Von Neumann es ligeramente más complicado. Por ejemplo, si se considera la ecuación hiperbólica de onda dimensional con velocidad constante v, se tiene

$$\frac{\partial^2 \mathbf{u}}{\partial t^2} = v^2 \frac{\partial^2 \mathbf{u}}{\partial x^2} \tag{9.29}$$

Reescribiendo la ecuación (9.29) como un conjunto de dos ecuaciones de primer orden, se llega a

$$\frac{\partial}{\partial t}\begin{bmatrix} r \\ s \end{bmatrix} = \frac{\partial}{\partial x}\begin{bmatrix} vs \\ vr \end{bmatrix} \tag{9.30}$$

El método de Lax para esta ecuación es

$$\begin{aligned} r_j^{n+1} &= \frac{1}{2}\left(r_{j+1}^n + r_{j-1}^n\right) + \frac{v\Delta t}{2\Delta x}\left(s_{j+1}^n - s_{j-1}^n\right) \\ s_j^{n+1} &= \frac{1}{2}\left(s_{j+1}^n + s_{j-1}^n\right) + \frac{v\Delta t}{2\Delta x}\left(r_{j+1}^n - r_{j-1}^n\right) \end{aligned} \tag{9.31}$$

El análisis de estabilidad supone que la solución tiene la siguiente forma:

$$\begin{bmatrix} r_j^n \\ s_j^n \end{bmatrix} = \xi^n e^{ikj\Delta x 0}\begin{bmatrix} r^0 \\ s^0 \end{bmatrix} \tag{9.32}$$

El vector del lado derecho de (9.32) es un vector propio, constante en espacio y tiempo, y ξ es un número complejo. Sustituyendo la ecuación (9.32) en la ecuación (9.31) y dividiendo entre ξ^n, se obtiene la ecuación vectorial homogénea,

$$\begin{bmatrix} \cos k\Delta x - \xi & i\dfrac{v\Delta t}{\Delta x}\operatorname{sen} k\Delta x \\ i\dfrac{v\Delta t}{\Delta x}\operatorname{sen} k\Delta x & \cos k\Delta x - \xi \end{bmatrix} \cdot \begin{bmatrix} r^0 \\ s^0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{9.33}$$

www.FreeLibros.me

La ecuación (9.33) admite la solución no nula si el determinante de la matriz del lado izquierdo se anula. Esto sucede con dos valores diferentes de ξ, los cuales se obtienen de la siguiente ecuación,

$$\xi = \cos k\Delta x \pm i \frac{v\Delta t}{\Delta x} \operatorname{sen} k\Delta x \tag{9.34}$$

La condición de estabilidad es que ambas soluciones satisfagan $|\xi| < 1$. Esto lleva a la condición de Courant-Friedrichs-Lewy.

9.2.3 Otras fuentes de error

Una variedad adicional de error es la evaluación de la exactitud. Por ejemplo, los esquemas de diferencias finitas para ecuaciones hiperbólicas pueden exhibir dispersión o introducir errores en la fase. Si se reescribe la ecuación (9.27) como

$$\xi = e^{-ik\Delta x} + i\left(1 - \frac{v\Delta t}{\Delta x}\right) \operatorname{sen} k\Delta x \tag{9.35}$$

Además, si se toma la superposición de valores propios con diferentes k como un conjunto inicial arbitrario de ondas solución, si $\Delta t = \frac{\Delta x}{v}$, se puede obtener la solución exacta de cada valor propio del conjunto de ondas multiplicándolos por $e^{-ik\Delta x}$. Sin embargo, si $\frac{v\Delta t}{\Delta x}$ no es exactamente uno, los valores propios dados por la ecuación (9.35) presentan dispersión. Ésta será tan grande como sea la longitud de onda en comparación con el espaciado de la malla Δx. El tercer tipo de error está asociado a la no linealidad de las ecuaciones; por tanto, se llama *inestabilidad no lineal*. Por ejemplo, si se toma una parte de la ecuación para el *flujo de fluidos de Euler*,

$$\frac{\partial v}{\partial t} = -v \frac{\partial v}{\partial x} + \dots, \tag{9.36}$$

el término no lineal en v hace que el contorno de la onda sea más pronunciado. Como el análisis de Von Neumann sugiere que la estabilidad depende de $k\Delta x$, un esquema estable para contornos suaves se vuelve así sensible para contornos pronunciados. En el esquema de Lax, en la ecuación (9.26), una perturbación u en el punto j se propaga a los puntos $j+1$ y $j-1$ en el siguiente intervalo de tiempo.

9.2.4 Diferenciador contraviento

En general, el diferenciador contraviento añade precisión a los problemas donde las variables experimentan cambios bruscos de condición; por ejemplo, cuando atraviesan una depresión o cualquier discontinuidad. Así, la forma más sencilla de modelar las propiedades de transporte es usar este diferenciador, el cual tiene la siguiente estructura:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -v_j^n \begin{cases} \dfrac{u_j^n - u_{j-1}^n}{\Delta x}, & v_j^n > 0 \\ \dfrac{u_{j+1}^n - u_j^n}{\Delta x}, & v_j^n < 0 \end{cases} \tag{9.37}$$

Este esquema se muestra gráficamente en la figura 9.7 y sólo es preciso calcular las derivadas espaciales de primer orden, debido a que la meta de una implementación numérica no es siempre la exactitud en el sentido matemático estricto, sino la fidelidad respecto a la representación física de un sistema.

Para el esquema diferenciador de la ecuación (9.37) el factor amplificador, en cada caso, con v constante es

$$\xi = 1 - \left|\frac{v\Delta t}{\Delta x}\right|(1 - \cos k\Delta x) - i \frac{v\Delta t}{\Delta x} \operatorname{sen} k\Delta x \tag{9.38}$$

www.FreeLibros.me

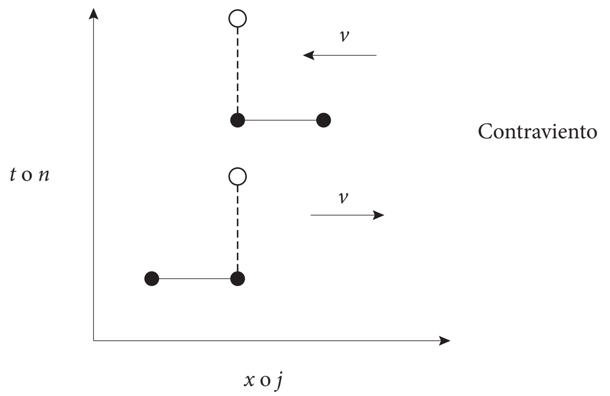


Figura 9.7 Representación de los esquemas diferenciadores contraviento. El esquema superior es estable cuando la constante de advección *v* es negativa, como se muestra. El esquema inferior es estable cuando la constante de advección es positiva.

o bien

$$|\xi|^2 = 1 - 2\left|\frac{v\Delta t}{\Delta x}\right|\left(1 - \left|\frac{v\Delta t}{\Delta x}\right|\right)(1 - \cos k\Delta x) \tag{9.39}$$

Por el criterio de estabilidad $|\xi|^2 \leq 1$ es, nuevamente, el criterio de Courant-Friedrichs-Lewy.

9.2.5 Precisión de segundo orden en tiempo

Si se usa un método que es preciso de primer orden en tiempo y de segundo orden en espacio, se toma $v\Delta t \leq \Delta x$ para obtener una buena precisión. Así, la condición de Courant-Friedrichs-Lewy no es factor limitante. Sin embargo, sí lo es para esquemas de segundo orden apropiados, tanto en tiempo como en espacio, de los cuales los más comunes son el método escalonado de salto de rana y el método de Lax-Wendroff. Ambos se detallan a continuación:

9.2.5.1 Método escalonado de salto de rana

En la figura 9.8 se muestra gráficamente el método escalonado de salto de rana para la ecuación de conservación (9.16). Éste se define matemáticamente como

$$u_j^{n+1} - u_j^{n-1} = -\frac{\Delta t}{\Delta x}\left(F_{j+1}^n - F_{j-1}^n\right) \tag{9.40}$$

En la ecuación (9.40), con los valores u^n en el tiempo t^n, se calculan los flujos F_j^n, y a continuación se calculan los nuevos valores u^{n+1} utilizando los valores centrales de tiempo. El método escalonado para la ecuación de flujo conservativo (9.17), toma la siguiente forma:

$$u_j^{n+1} - u_j^{n-1} = -v\frac{\Delta t}{\Delta x}\left(u_{j+1}^n - u_{j-1}^n\right) \tag{9.41}$$

El análisis de estabilidad de Von Neumann da una ecuación cuadrática para ξ, debido a que al sustituir la ecuación de un valor propio en la ecuación (9.41), se obtienen dos potencias consecutivas de ξ,

$$\xi^2 - 1 = -2i\xi\frac{v\Delta t}{\Delta x}\operatorname{sen} k\Delta x \tag{9.42}$$

Cuya solución es

$$\xi = -i\frac{v\Delta t}{\Delta x}\operatorname{sen} k\Delta x \pm \sqrt{1 - \left(\frac{v\Delta t}{\Delta x}\operatorname{sen} k\Delta x\right)^2} \tag{9.43}$$

www.FreeLibros.me

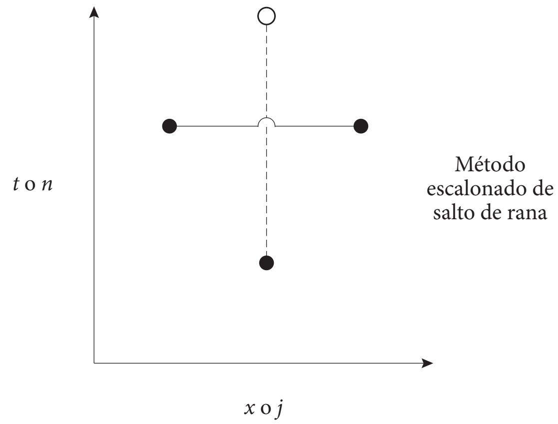


Figura 9.8 Representación del esquema diferenciador del salto de rana escalonado. Este esquema es muy preciso, de segundo orden en espacio y tiempo.

Así, para la estabilidad se requiere del cumplimiento de la condición de Courant-Friedrichs-Lewy. De hecho, en la ecuación (9.43), el cuadrado del valor absoluto de ξ es igual a uno, para un valor de $v\Delta t \leq \Delta x$; por tanto, no hay disipación de amplitud.

La diferenciación de las ecuaciones del método escalonado de salto de rana para la ecuación (9.30), por convención de notación, es más adecuada si las variables están centradas sobre los puntos de media malla; es decir, si se tiene

$$\begin{aligned} r_{j+\frac{1}{2}}^{n} &\equiv v \left.\frac{\partial u}{\partial x}\right|_{j+\frac{1}{2}}^{n} = v \frac{u_{j+1}^{n} - u_{j}^{n}}{\Delta x} \\ s_{j}^{n+\frac{1}{2}} &\equiv \left.\frac{\partial u}{\partial t}\right|_{j}^{n+\frac{1}{2}} = \frac{u_{j}^{n+1} - u_{j}^{n}}{\Delta t} \end{aligned} \tag{9.44}$$

Pero si en la malla se tienen definidos r y s, el diferenciador de salto de rana de la ecuación (9.30) es

$$\begin{aligned} \frac{r_{j+\frac{1}{2}}^{n+1} - r_{j+\frac{1}{2}}^{n}}{\Delta t} &= v \frac{s_{j+1}^{n+\frac{1}{2}} - s_{j}^{n+\frac{1}{2}}}{\Delta x} \\ \frac{s_{j}^{n+\frac{1}{2}} - s_{j}^{n-\frac{1}{2}}}{\Delta t} &= v \frac{r_{j+\frac{1}{2}}^{n} - r_{j-\frac{1}{2}}^{n}}{\Delta x} \end{aligned} \tag{9.45}$$

Si se sustituye la ecuación (9.32) en la ecuación (9.45), se llega nuevamente a la condición de Courant-Friedrichs-Lewy como una necesidad para la estabilidad, y cuando se cumple, se tiene la característica de no disipación de amplitud. Si se sustituye la ecuación (9.44) en la (9.45), ésta equivale a

$$\frac{u_{j}^{n+1} - 2u_{j}^{n} + u_{j}^{n-1}}{(\Delta t)^{2}} = v^{2} \frac{u_{j+1}^{n} - 2u_{j}^{n} + u_{j-1}^{n}}{(\Delta x)^{2}} \tag{9.46}$$

Este esquema es de dos niveles: son necesarias u^{n} y u^{n-1} para obtener u^{n+1}. El método de salto de rana es inestable para ecuaciones no lineales cuando los gradientes son grandes. La inestabilidad se relaciona con el hecho de que los puntos pares e impares de la malla están completamente desacoplados (véase la figura 9.9). Esta inestabilidad se puede remediar acoplando las dos mallas por medio de un término de viscosidad, es decir, sumando $\left(-2u_{j}^{n} + 2u_{j-1}^{n}\right)$, un pequeño coeficiente menor que uno en el lado derecho de la ecuación (9.41). Así se obtiene

$$u_{j}^{n+1} - u_{j}^{n-1} = -v \frac{\Delta t}{\Delta x}\left(u_{j+1}^{n} - 2u_{j}^{n} + u_{j-1}^{n}\right) \tag{9.47}$$

www.FreeLibros.me

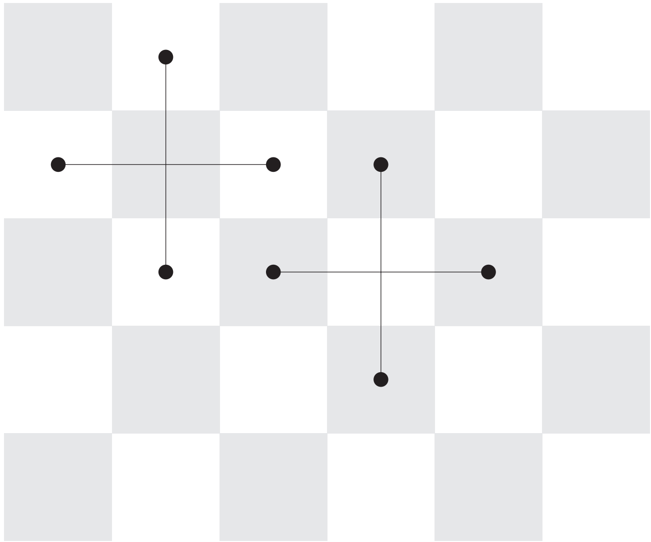


Figura 9.9 El origen de la inestabilidad es un esquema de salto de rana. Si tomamos a los puntos de la malla como cuadrados de un tablero de ajedrez, entonces los cuadrados blancos se acoplan entre sí, y los negros también; pero no hay acoplamiento entre negros y blancos. La estrategia aquí, es introducir una pequeña pieza de difusión que acople la malla.

9.2.5.2 Método de Lax-Wendroff

El esquema de dos pasos de Lax-Wendroff es un método de segundo orden en tiempo que evita la disipación numérica, ya que se pueden definir valores intermedios $u_{j+\frac{1}{2}}$ en los medios pasos de tiempo $t_{j+\frac{1}{2}}$. Los puntos medios de la malla $x_{j+\frac{1}{2}}$ se calculan con el esquema de Lax, como

$$u_{j+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{1}{2}\left(u_{j+1}^n + u_j^n\right) - \frac{\Delta t}{2\Delta x}\left(F_{j+1}^n + F_j^n\right) \tag{9.48}$$

Con estos valores se calculan los flujos $F_{j+\frac{1}{2}}^{n+\frac{1}{2}}$. Entonces los valores actuales u_j^{n+1} se calculan con la expresión

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}\left(F_{j+\frac{1}{2}}^{n+\frac{1}{2}} - F_{j-\frac{1}{2}}^{n+\frac{1}{2}}\right) \tag{9.49}$$

Los valores provisionales $u_{j+\frac{1}{2}}^{n+\frac{1}{2}}$ se descartan, como se muestra gráficamente en la figura 9.10. Sustituyendo la ecuación (9.48) en la ecuación (9.49) se obtiene

$$u_j^{n+1} = u_j^n - \alpha\left[\frac{1}{2}\left(u_{j+1}^n + u_j^n\right) - \frac{1}{2}\alpha\left(u_{j+1}^n + u_j^n\right) - \frac{1}{2}\left(u_j^n + u_{j-1}^n\right) + \frac{1}{2}\alpha\left(u_j^n - u_{j-1}^n\right)\right] \tag{9.50}$$

donde

$$\alpha \equiv \frac{v\Delta t}{\Delta x} \tag{9.51}$$

Entonces,

$$\xi = 1 - i\alpha \operatorname{sen} k\Delta x - \alpha^2\left(1 - \cos k\Delta x\right) \tag{9.52a}$$

y

$$|\xi|^2 = 1 - \alpha^2\left(1 - \alpha^2\right)\left(1 - \cos k\Delta x\right)^2 \tag{9.52b}$$

www.FreeLibros.me

El criterio de estabilidad $|\xi|^2 \leq 1$ es, por tanto, $\alpha^2 \leq 1$; por tanto, de la ecuación (9.51) se establece la relación $v\Delta t \leq \Delta x$ como criterio de estabilidad.

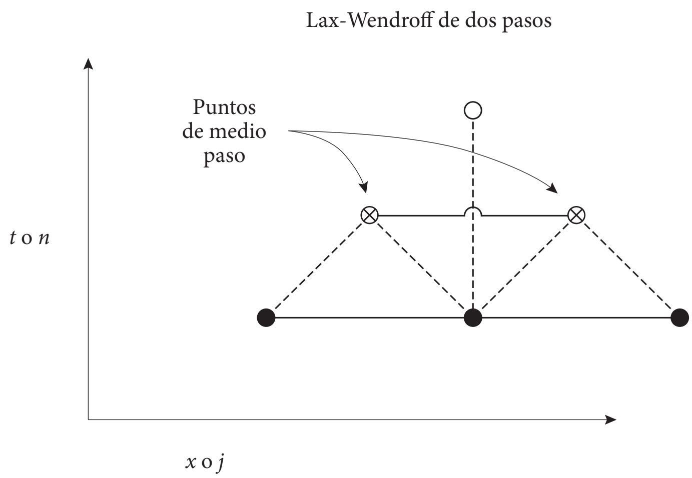


Figura 9.10 Representación del esquema diferenciador Lax-Wendroff de dos pasos. Se calculan dos pasos medios con el método Lax. Éstos, más uno de los puntos originales, producen el punto nuevo a través del salto de rana escalonado.

9.3 Problemas de valor inicial difusos

La ecuación típica de difusión en una dimensión es la ecuación parabólica, dada por

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(D\frac{\partial u}{\partial x}\right), \tag{9.53}$$

donde D es el coeficiente de difusión $(D \geq 0)$. Para el caso donde D es constante, se llega a

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} \tag{9.54}$$

La ecuación (9.54) se puede diferenciar numéricamente como

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = D\left[\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2}\right] \tag{9.55}$$

Este esquema FTCS es inestable para una ecuación hiperbólica; sin embargo, el factor de amplificación de la ecuación (9.55) es

$$\xi = 1 - \frac{4D\Delta t}{(\Delta x)^2}\operatorname{sen}^2\left(\frac{k\Delta x}{2}\right) \tag{9.56}$$

El requerimiento de $|\xi| \leq 1$ lleva al criterio de estabilidad,

$$\frac{2D\Delta t}{(\Delta x)^2} \leq 1 \tag{9.57}$$

La interpretación física de esta restricción es el máximo paso de tiempo permitido más allá del tiempo de difusión a través de una célula de ancho Δx. En general, el tiempo τ a través de una escala espacial de tamaño λ es del orden

www.FreeLibros.me

$$\tau \prec \frac{\lambda^2}{D} \tag{9.58}$$

9.3.1 Método de Crank-Nicolson

Éste es un método exacto de segundo orden en tiempo. Este esquema se basa en el esquema de la ecuación (9.54), excepto que las derivadas espaciales se evalúan en los pasos de tiempo $n+1$. Así, el esquema queda de la siguiente manera:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = D\left[\frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{(\Delta x)^2}\right] \tag{9.59}$$

Los esquemas de este tipo se llaman de *tiempo completamente implícito* o *retrasado*. Para resolver la ecuación (9.59), se tienen que despejar los u_j^{n+1} en cada paso de tiempo de un conjunto de ecuaciones simultáneas de primer grado. de la forma

$$-\alpha u_{j-1}^{n+1} + (1+2\alpha)u_j^{n+1} - \alpha u_{j+1}^{n+1} = u_j^n \qquad j = 1, 2, 3, \ldots, J \tag{9.60}$$

donde

$$\alpha \equiv \frac{D\Delta t}{(\Delta x)^2}$$

El factor de amplificación de la ecuación (9.60) es

$$\xi = \frac{1}{1 + 4\alpha \operatorname{sen}^2\left(\frac{k\Delta x}{2}\right)} \tag{9.61}$$

Evidentemente, $|\xi| < 1$ para algún tamaño de paso Δt. Así, el esquema es incondicionalmente estable; ésta es la característica más destacada de los métodos implícitos. Así, combinando la estabilidad de un método implícito con la exactitud de un método explícito de segundo orden en espacio y tiempo, con sólo tomar el promedio de ambos, se llega a

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \frac{D}{2}\left[\frac{\left(u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}\right) + \left(u_{j+1}^n - 2u_j^n + u_{j-1}^n\right)}{(\Delta x)^2}\right] \tag{9.62}$$

El factor de amplificación es

$$\xi = \frac{1 - 2\alpha \operatorname{sen}^2\left(k\Delta \frac{x}{2}\right)}{1 + 2\alpha \operatorname{sen}^2\left(k\Delta \frac{x}{2}\right)} \tag{9.63}$$

Por lo que el método es estable para cualquier paso Δt o Δx. La figura 9.11 esquematiza el método. En la figura, 1), FTCS, tiene precisión de primer orden, pero es estable sólo para tamaños de pasos de tiempo lo bastante pequeños; 2), totalmente implícito, es estable para pasos de tiempo arbitrariamente grandes; pero sólo tiene precisión de primer orden; 3), Crank-Nicolson, tiene precisión de segundo orden y, en general, es estable para tamaños de paso de tiempo grandes.

www.FreeLibros.me

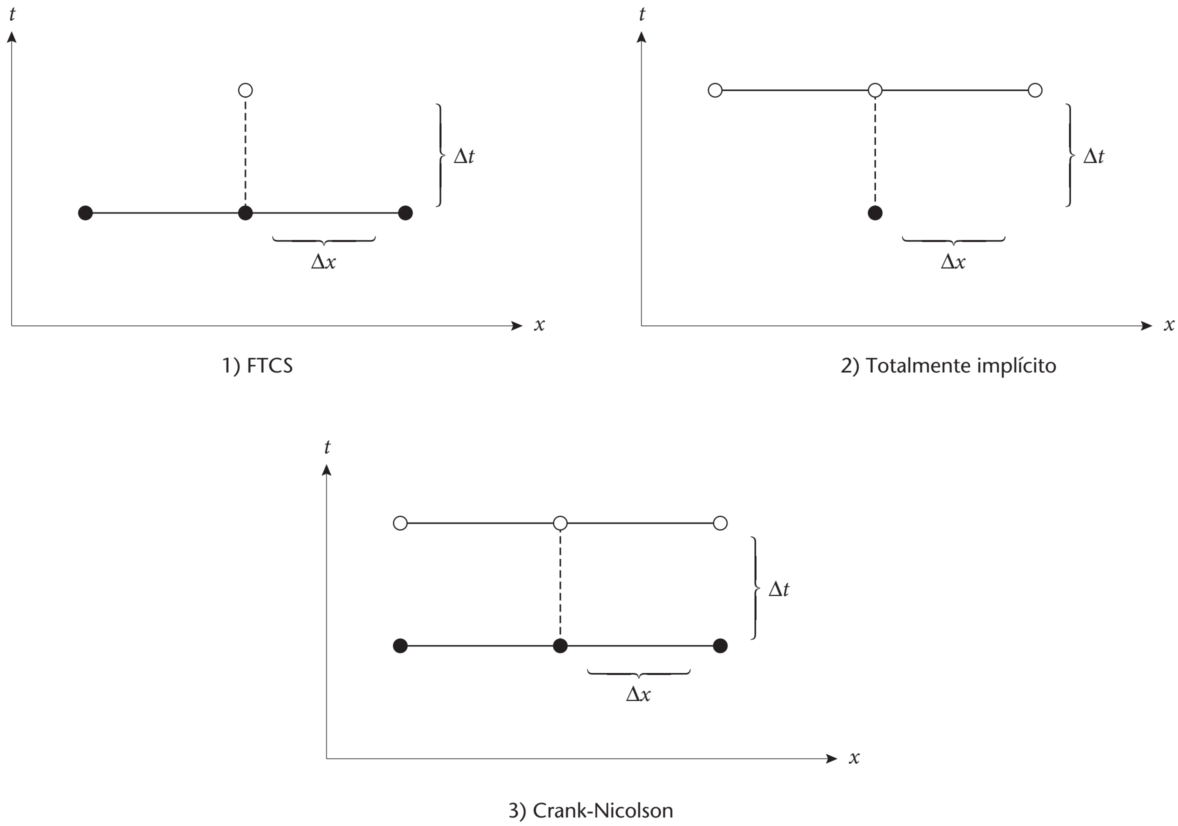


Figura 9.11 Tres esquemas de diferenciación para problemas de difusión.

9.3.1.1 Primera generalización

Suponiendo que el coeficiente D no es constante en la ecuación de difusión, por ejemplo, $D = D(x)$, primero se adapta un cambio de variable simplemente analítico para obtener

$$y = \int \frac{dx}{D(x)} \tag{9.64}$$

Entonces,

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} D(x) \frac{\partial u}{\partial x} \tag{9.65}$$

Esto conduce a

$$\frac{\partial u}{\partial t} = \frac{1}{D(y)} \frac{\partial^2 u}{\partial y^2} \tag{9.66}$$

Evaluando D y la y_i apropiada heurísticamente, el criterio de estabilidad dado por la ecuación (9.58) sería

$$\Delta t \leq \min_j \left[\frac{(\Delta y)^2}{2D_j} \right] \tag{9.67}$$

La ecuación anterior denota que el espaciamiento constante Δy en y no implica un espaciamiento constante Δt en t.

www.FreeLibros.me

De manera alterna, una estrategia que no necesita formas analíticas modificables de D es, simplemente, la ecuación de diferencias centrales de (9.65), de donde se obtiene

$$\frac{u_j^{n+1}-u_j^n}{\Delta t}=\frac{D_{j+\frac{1}{2}}\left(u_{j+1}^n-u_j^n\right)-D_{j-\frac{1}{2}}\left(u_j^n-u_{j-1}^n\right)}{\left(\Delta x\right)^2} \tag{9.68}$$

donde $D_{j+\frac{1}{2}} \equiv D\left(x_{j+\frac{1}{2}}\right)$, y el criterio de estabilidad heurístico es

$$\Delta t \le \min_j \left[\frac{\left(\Delta y\right)^2}{2D_{j+\frac{1}{2}}}\right] \tag{9.69}$$

9.3.1.2 Segunda generalización

Si se tiene un problema de difusión no lineal, por ejemplo con $D=D(u)$, y si se tiene un esquema explícito, éste se puede generalizar de manera simple. Por ejemplo, en la ecuación (9.68) se tendría

$$D_{j+\frac{1}{2}}=\frac{1}{2}\left[D\left(u_{j+1}^n\right)+D\left(u_j^n\right)\right] \tag{9.70}$$

En cambio, un esquema implícito, reemplazando n por $n+1$ en la ecuación (9.70), conduce a un conjunto de ecuaciones no lineales por resolver en cada paso de tiempo.

Otra alternativa es si la forma $D=D(u)$ lleva a integrar

$$dz=D(u)du \tag{9.71}$$

analíticamente, para $z(u)$. Entonces, el lado derecho de la ecuación (9.53) sería

$$\frac{\partial^2 z}{\partial x^2} \tag{9.72}$$

diferenciando explícitamente la expresión (9.72), se llega a

$$\frac{z_{j+1}^{n+1}-2z_j^{n+1}+z_{j-1}^{n+1}}{\left(\Delta x\right)^2} \tag{9.73}$$

Linealizando cada término de la expresión (9.73), se obtiene por ejemplo

$$z_j^{n+1} \equiv z\left(u_j^{n+1}\right)=z\left(u_j^n\right)+\left(u_j^{n+1}-u_j^n\right)\frac{\partial z}{\partial u}\bigg|_{j,n} \tag{9.74}$$

es decir,

$$z\left(u_j^{n+1}\right)=z\left(u_j^n\right)+\left(u_j^{n+1}-u_j^n\right)D\left(u_j^n\right) \tag{9.75}$$

Esto reduce el problema a una forma tridiagonal, y se mantienen las ventajas de la estabilidad de diferenciación totalmente implícita.

9.3.2 Ecuación de Schrödinger

La ecuación de Schrödinger en mecánica cuántica es, básicamente, una ecuación parabólica. Para el caso específico de considerar la difusión de un grupo de ondas mediante un potencial $V(x)$, la ecuación toma la forma

www.FreeLibros.me

$$i\frac{\partial \psi}{\partial t}=\frac{\partial^2 \psi}{\partial x^2}+V(x)\psi \quad \textbf{(9.76)}$$

La condición inicial $\psi(x,t=0)$ y las condiciones de frontera, $\psi=0$ si $t\to\pm\infty$, con un esquema implícito estable, llevan a la generalización de la ecuación (9.76) de la forma

$$i\left[\frac{\psi_j^{n+1}-\psi_j^n}{\Delta t}\right]=-\left[\frac{\psi_{j+1}^{n+1}-2\psi_j^{n+1}+\psi_{j-1}^{n+1}}{(\Delta x)^2}\right]+V_j\psi_j^{n+1} \quad \textbf{(9.77)}$$

para el cual

$$\xi=\frac{1}{1+i\left[\frac{4\Delta t}{(\Delta x)^2}\operatorname{sen}^2\left(\frac{k\Delta x}{2}\right)+V_j\Delta t\right]} \quad \textbf{(9.78)}$$

El esquema es incondicionalmente estable, pero no unitario, y el problema requiere, desde el punto de vista físico, que la probabilidad total de encontrar una partícula en cualquier lugar sea la unidad; por último, esto se puede expresar como

$$\int_{-\infty}^{\infty}|\psi|^2\,dx=1 \quad \textbf{(9.79)}$$

La función de onda inicial $\psi(x,t=0)$ se normaliza para satisfacer (9.79). La ecuación de Schrödinger (9.76) garantiza que esta condición se satisface en todos los tiempos posteriores. Entonces, si se escribe la ecuación de Schrödinger de la forma

$$i\frac{\partial \psi}{\partial t}=H\psi, \quad \textbf{(9.80)}$$

donde

$$H=-\frac{\partial^2}{\partial x^2}+V(x),$$

la expresión formal de la ecuación es

$$\psi(x,t)=e^{-iHt}\psi(x,0) \quad \textbf{(9.81)}$$

El esquema explícito FTCS se aproxima a aproxima (9.81) mediante

$$\psi_j^{n+1}=(1-iH\Delta t)\psi_j^n \quad \textbf{(9.82)}$$

El esquema implícito en contraste es

$$\psi_j^{n+1}=(1-iH\Delta t)^{-1}\psi_j^n \quad \textbf{(9.83)}$$

Ambos tienen precisión de primer orden. Sin embargo, ninguno de los operadores en las ecuaciones (9.82) y (9.83) es unitario.

La forma correcta de diferenciar la ecuación de Schrödinger es utilizando la forma de Cayley para la representación de diferencias finitas de e^{-iHt}, la cual tiene una precisión de segundo orden, es unitaria y está dada por

$$e^{-iHt}\cong\frac{1-\frac{1}{2}iH\Delta t}{1+\frac{1}{2}iH\Delta t} \quad \textbf{(9.84)}$$

En otras palabras,

$$\left(1+\frac{1}{2}iH\Delta t\right)\psi_j^{n+1}=\left(1-\frac{1}{2}iH\Delta t\right)\psi_j^n \quad \textbf{(9.85)}$$

www.FreeLibros.me

Al reemplazar H por su aproximación de diferencias finitas en x, se llega a un sistema tridiagonal; de hecho se llega por este medio al método de Crank-Nicolson.

9.4 Problemas de valor en la frontera

La mayoría de los problemas con valores en la frontera se reducen a resolver un sistema de ecuaciones lineales disperso, de la forma

$$\mathbf{Au} = \mathbf{b} \tag{9.86}$$

Estos sistemas son el resultado de la solución de ecuaciones elípticas. El sistema (9.86) se resuelve en forma directa si es lineal, o iterativamente para ecuaciones con valor en la frontera que son no lineales. Si el sistema (9.86) es lineal con coeficientes constantes en el espacio, una técnica de solución rápida es el método de la transformada de Fourier. Un método más general es la *reducción cíclica*. Ambos exigen que la frontera coincida con los ejes coordenados. Finalmente, para algunos problemas, la combinación de ambos métodos es la más adecuada.

9.4.1 Método de la transformada de Fourier

La transformada inversa de Fourier en dos dimensiones es

$$u_{ij} = \frac{1}{JL}\sum_{m=0}^{J-1}\sum_{n=0}^{L-1}\hat{u}_{mn} e^{\frac{-2\pi ijm}{J}} e^{\frac{-2n\pi il}{L}} \tag{9.87}$$

Esta ecuación se puede resolver con el algoritmo de la transformada rápida de Fourier (FFT) de manera independiente en cada dimensión. En forma similar, se tiene también

$$\rho_{ij} = \frac{1}{JL}\sum_{m=0}^{J-1}\sum_{n=0}^{L-1}\hat{\rho}_{mn} e^{\frac{-2\pi ijm}{J}} e^{\frac{-2n\pi il}{L}} \tag{9.88}$$

Considerando la representación en diferencias finitas de la ecuación elíptica de Poisson (ecuación 9.6), ésta es

$$u_{j+1,\,l} + u_{j-1,\,l} + u_{j,\,l+1} + u_{j,\,l-1} - 4u_{j,\,l} = \Delta^2 \rho_{j,\,l}$$

Sustituyendo las expresiones (9.87) y (9.88) en la ecuación anterior, se obtiene

$$\hat{u}_{mn}\left(e^{\frac{2\pi im}{J}} + e^{\frac{-2\pi im}{J}} + e^{\frac{2\pi in}{J}} + e^{\frac{-2\pi in}{J}} - 4\right) = \hat{\rho}_{mn}\Delta^2 \tag{9.89}$$

Despejando $\hat{u}_{mn}$ de la ecuación (9.89) y utilizando la equivalencia de Euler para representar las funciones exponenciales como cosenos se tiene

$$\hat{u}_{mn} = \frac{\hat{\rho}_{mn}\Delta^2}{2\left(\cos\frac{2\pi m}{J} + \cos\frac{2\pi n}{L} - 2\right)} \tag{9.90}$$

y la transformada de Fourier de $\hat{\rho}_{mn}$ es

$$\hat{\rho}_{mn} = \sum_{j=0}^{J-1}\sum_{l=0}^{L-1}\rho_{il} e^{\frac{2\pi imj}{J}} e^{\frac{2\pi inl}{L}} \tag{9.91}$$

Así, en forma resumida, la estrategia para resolver la ecuación elíptica de Poisson utilizando la FFT es

- Calcular $\hat{\rho}_{mn}$ de la ecuación (9.91),

www.FreeLibros.me

- Calcular $\hat{u}_{mn}$ de la ecuación (9.90),
- Calcular u_{jl} con la transformada inversa de Fourier dada por la ecuación (9.87).

El procedimiento anterior es válido para condiciones periódicas de frontera. En otras palabras, la solución satisface

$$u_{jl} = u_{j+J,\,l} = u_{j,\,l+L}$$

9.4.2 Condiciones de frontera de Dirichlet

Si se considera la condición de frontera de Dirichlet $u = 0$, sobre la frontera rectangular, en lugar de la expansión, dada por la ecuación (9.87) se utiliza la expansión en ondas seno, dadas por

$$u_{il} = \frac{2}{J}\frac{2}{L}\sum_{m=1}^{J-1}\sum_{n=1}^{L-1}\hat{u}_{mn}\,\text{sen}\frac{\pi jm}{J}\,\text{sen}\frac{\pi nl}{L} \tag{9.92}$$

La ecuación (9.92) satisface la condición de frontera de $u = 0$ en $j = 0, J$ y en $l = 0, L$. Si se sustituye esta expansión y su análoga para ρ_{il} en la ecuación de Poisson, dada por (9.6), entonces se obtiene un procedimiento paralelo al de las condiciones periódicas en la frontera. Este procedimiento es como sigue:

- Se calcula $\hat{\rho}_{mn}$ mediante la transformada seno,

$$\hat{\rho}_{mn} = \sum_{j=1}^{J-1}\sum_{l=1}^{L-1}\rho_{jl}\,\text{sen}\frac{\pi jm}{J}\,\text{sen}\frac{\pi nl}{L}$$

- Se calcula $\hat{u}_{mn}$ por la expresión

$$\hat{u}_{mn} = \frac{\Delta^2\hat{\rho}_{mn}}{2\left(\cos\frac{\pi m}{J} + \cos\frac{\pi n}{L} - 2\right)}$$

- Se calcula u_{jl} mediante la transformada inversa seno dada por (9.92).

9.4.3 Condiciones de frontera no homogéneas

Si se tienen condiciones en la frontera no homogéneas, por ejemplo, $u = 0$ en todas las fronteras, excepto en $u = f(y)$ en la frontera $x = J\Delta$, se tiene que agregar a la solución anterior una solución u^H de la ecuación

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \tag{9.93}$$

Esto satisface las condiciones de frontera requeridas. En el caso continuo, se tiene la expansión de la forma

$$u^H = \sum_n A_n\,\text{senh}\frac{n\pi x}{L\Delta}\,\text{sen}\frac{n\pi y}{L\Delta} \tag{9.94}$$

donde A_n se encuentra estableciendo la restricción $u = f(y)$ en $x = J\Delta$.

Así, en forma discreta, se tiene,

$$u_{jl}{}^H = \frac{2}{L}\sum_{n=1}^{L-1}A_n\,\text{senh}\frac{n\pi j}{L}\,\text{sen}\frac{n\pi l}{L} \tag{9.95}$$

www.FreeLibros.me

Si $f(y=l\Delta)\equiv f_l$, se obtiene A_n de la fórmula inversa

$$A_n=\frac{1}{\operatorname{senh}(\pi nJ/L)}\sum_{l=1}^{L-1} f_l \operatorname{sen}\frac{\pi nl}{L} \quad \textbf{(9.96)}$$

Así, la solución completa es

$$u=u_{jl}+u_{jl}^{H} \quad \textbf{(9.97)}$$

Un procedimiento más sencillo para manejar términos no homogéneos es modificar la ecuación (9.6); así, el término de la fuente efectiva será ρ_{il} más una contribución de los términos de frontera. Esta ecuación queda de la siguiente forma:

$$\begin{aligned} &u'_{j+1,\,l}+u'_{j-1,\,l}+u'_{j,\,l+1}+u'_{j,\,l-1}-4u'_{j,\,l}= \\ &-\left(u^B_{j+1,\,l}+u^B_{j-1,\,l}+u^B_{j,\,l+1}+u^B_{j,\,l-1}-4u^B_{j,\,l}\right)+\Delta^2\rho_{j,\,l} \end{aligned} \quad \textbf{(9.98)}$$

En la ecuación (9.98), los términos u^B son de la forma

$$u^B_{J,\,l}=f_l \quad \textbf{(9.99)}$$

Todos los términos u^B de la ecuación (9.98) desaparecen, excepto cuando la ecuación se evalúa en $j=J-1$, dando

$$u'_{J,\,l}+u'_{J-2,\,l}+u'_{J-1,\,l+1}+u'_{J-1,\,l-1}-4u'_{J-1,\,l}=-f_l+\Delta^2\rho_{J-1,\,l} \quad \textbf{(9.100)}$$

El problema ahora equivale al caso de condiciones de frontera cero, sólo que un término fila de la fuente se modifica por el reemplazo,

$$\Delta^2\rho_{J-1,\,l}\rightarrow\Delta^2\rho_{J-1,\,l}-f_l \quad \textbf{(9.101)}$$

9.4.4 Condiciones de frontera de Neumann

En el caso de las condiciones de frontera de Neumann $\nabla u=0$, se manejan por la expansión coseno dada por

$$u_{jl}=\frac{2}{J}\frac{2}{L}\sum_{m=0}^{J}{}''\sum_{n=0}^{L}{}''\hat{u}_{mn}\cos\frac{\pi jm}{J}\cos\frac{l\pi n}{L} \quad \textbf{(9.102)}$$

La notación doble prima significa que los términos para $m=0$ y $m=J$ están divididos entre dos; similarmente para $n=0$ y $n=L$. Las condiciones no homogéneas $\nabla u=g$ se incluyen como se describió anteriormente. Por ejemplo, la condición

$$\frac{\partial u}{\partial x}=g(y) \text{ para } x=0 \quad \textbf{(9.103)}$$

pasa a su forma discreta como

$$\frac{u_{1,\,l}-u_{-1,\,l}}{2\Delta}=g_l \quad \textbf{(9.104)}$$

donde $g_l\equiv g(y=l\Delta)$.

∇u^B toma el valor prescrito en la frontera, pero u^B se anula excepto en el extremo de la frontera. Así, de la ecuación (9.104) se obtiene

$$u^B_{-1,\,l}=-2\Delta g_l \quad \textbf{(9.105)}$$

www.FreeLibros.me

Todos los términos de u^B desaparecen, excepto cuando $j=0$. Por tanto, la ecuación (9.98) se transforma en

$$u'_{1,\,l}+u'_{-1,\,l}+u'_{0,\,l+1}+u'_{0,\,l-1}-4u'_{0,\,l}=2\Delta g_l+\Delta^2\rho_{0,\,l} \tag{9.106}$$

Por tanto, u' es la solución de un problema de gradiente cero, con el término de la fuente modificado por el reemplazo,

$$\Delta^2\rho_{0,\,l}\rightarrow\Delta^2\rho_{0,\,l}+2\Delta g_l \tag{9.107}$$

9.4.5 Reducción cíclica

La transformada de Fourier es aplicable sólo en el caso de ecuaciones diferenciales parciales con coeficientes constantes. Para el caso general, se puede usar el método de reducción cíclica, que se ilustra aplicándolo a la *ecuación de Helmholtz*

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+b(y)\frac{\partial u}{\partial y}+c(y)u=g(x,y) \tag{9.108}$$

En su forma vectorial, la ecuación (9.108) en diferencias finitas se representa por

$$\mathbf{u}_{j-1}+\mathbf{T}\cdot\ \mathbf{u}_j+\mathbf{u}_{j+1}=\mathbf{g}_j\Delta^2 \tag{9.109}$$

donde $\mathbf{T}=\mathbf{B}-2\mathbf{l}$; el método de reducción cíclica parte de tres ecuaciones sucesivas de la forma de la ecuación (9.109). Así se tiene que

$$\begin{aligned}\mathbf{u}_{j-2}+\mathbf{T}\cdot\ \mathbf{u}_{j-1}+\mathbf{u}_j&=\mathbf{g}_{j-1}\Delta^2\\ \mathbf{u}_{j-1}+\mathbf{T}\cdot\ \mathbf{u}_j+\mathbf{u}_{j+1}&=\mathbf{g}_j\Delta^2\\ \mathbf{u}_j+\mathbf{T}\cdot\ \mathbf{u}_{j+1}+\mathbf{u}_{j+2}&=\mathbf{g}_{j+1}\Delta^2\end{aligned} \tag{9.110a, b, c}$$

Multiplicando la ecuación (9.110b) por $-\mathbf{T}$ y sumando las tres ecuaciones, se llega a

$$\mathbf{u}_{j-2}+\mathbf{T}^{(1)}\cdot\ \mathbf{u}_j+\mathbf{u}_{j+2}=\mathbf{g}_j^{(1)}\Delta^2 \tag{9.111}$$

Esta ecuación tiene la misma estructura que la ecuación (9.109), pero con

$$\begin{aligned}\mathbf{T}^{(1)}&=2\mathbf{l}-\mathbf{T}^2\\ \mathbf{g}_j^{(1)}&=\Delta^2(\mathbf{g}_{j-1}-\mathbf{T}\cdot\ \mathbf{g}_j+\mathbf{g}_{j+1})\end{aligned}$$

Como las ecuaciones resultantes tienen la misma forma que la original, se puede repetir el proceso hasta obtener una sola ecuación de la forma

$$\mathbf{T}^{(f)}\cdot\ \mathbf{u}_{J/2}=\Delta^2\mathbf{g}_{J/2}^{(f)}-\mathbf{u}_0-\mathbf{u}_J \tag{9.112}$$

En esta ecuación, las condiciones de frontera son $\mathbf{u}_0$ y $\mathbf{u}_J$. La solución buscada $\mathbf{u}_{J/2}$ queda entonces en función de las condiciones de frontera.

9.4.6 Reducción cíclica y análisis de Fourier

La mejor manera de resolver ecuaciones de la forma (9.108), incluyendo el caso de coeficientes constantes, es una combinación de análisis de Fourier y reducción cíclica. Si en la r-ésima etapa de reducción cíclica se realiza un análisis de Fourier, por ejemplo a la ecuación (9.111) a lo largo de (y), esto daría un sistema tridiagonal en dirección (x) para cada modo (y) de Fourier, de la forma

www.FreeLibros.me

$$\hat{u}^k_{j-2^r} + \lambda_k^{(r)} \hat{u}^k_j + \hat{u}^k_{j+2^r} = \Delta^2 g_j^{(r)k} \quad \textbf{(9.113)}$$

Aquí, $\lambda_k^{(r)}$ es el valor propio de $T^{(r)}$ correspondiente al k-ésimo modo de Fourier. El número de niveles adecuados de reducción cíclica para reducir al mínimo el número total de operaciones está dado en forma heurística como nivel óptimo, $r \to \log_2(\log_2 J)$, asintóticamente.

Por ejemplo, si se tiene un caso típico de acoplamiento de 128 × 128, se obtiene $r = \log_2(\log_2 128) = 1.5794$ asintóticamente. Por tanto, el nivel óptimo de reducción cíclica para este caso específico es $r = 2$.

Problemas propuestos

9.5.1 Implemente numéricamente, por el método FTCS, la siguiente ecuación diferencial parcial tipo hiperbólico:

$$-\frac{\partial i(x,t)}{\partial x} = C(x)\frac{\partial v(x,t)}{\partial t}$$

donde $C(x) = 4e^{-2x}$.

9.5.2 Implemente numéricamente, por el método escalonado de salto de rana, la siguiente ecuación diferencial parcial tipo hiperbólico

$$-\frac{\partial v(x,t)}{\partial x} = R(x)i(x,t) + L(x)\frac{\partial i(x,t)}{\partial t}$$

donde $R(x) = x$ y $L(x) = e^{0.01x}$.

9.5.3 Implemente numéricamente, por el método de Lax Wendroff, la siguiente ecuación diferencial parcial tipo hiperbólico

$$\frac{\partial v(x,t)}{\partial x} = -L\frac{\partial i(x,t)}{\partial t}$$

donde $L = 3$.

www.FreeLibros.me

www.FreeLibros.me

Respuestas a los problemas propuestos

Capítulo 1

1.6.1 La serie de Taylor en $x_0 = 0$ de $f(x) = x^4 - 3x^2 + 2$ coincide.

1.6.2 $p_4(x) = 1 + \frac{1}{2}x - \frac{1}{8}x^2 + \frac{1}{16}x^3 - \frac{5}{128}x^4$, $p_4(2) = 1.375$, $Error = 3.9214 \times 10^{-2}$.

1.6.3 $p_n(-x^2) = 1 - \frac{1}{1!}x^2 + \frac{1}{2!}x^4 - \frac{1}{3!}x^6 + \cdots + (-1)^n \frac{1}{n!}x^{2n}$

$$\int_0^1 e^{-x^2} dx = 0.7468241328$$

$$\int_0^1 e^{-x^2} dx \approx \int_0^1 \left(1 - \frac{1}{1!}x^2 + \frac{1}{2!}x^4 - \frac{1}{3!}x^6 + \frac{1}{8!}x^8\right) dx = \frac{5\,651}{7\,560} = 0.7474867725$$

$$Error = -6.\,6264 \times 10^4$$

1.6.4 $p_{11}(x) = x - \frac{1}{3}x^3 + \frac{1}{5}x^5 - \frac{1}{7}x^7 + \frac{1}{9}x^9 - \frac{1}{11}x^{11}$, dado que $\arctan(1) = \frac{\pi}{4}$ se tiene que $\frac{\pi}{4} = \arctan(1) \approx 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} = \frac{2\,578}{3\,465} = 0.744011544$. Entonces

$\pi \approx 2.976046176$, $Error = 0.1655464776$

1.6.5 Trabajando con una precisión de 10 decimales se tiene

1. $$E = p - \overline{p} = 3.1415926536 - 3.1416 = -0.73464 \times 10^{-5}$$
   $$E_r = \frac{p - \overline{p}}{P} = \frac{3.1415926536 - 3.1416}{3.1415926536} = -0.23384 \times 10^{-5}$$

2. $$E = p - \overline{p} = 3.1415926536 - 3.1428571429 = -0.001264$$
   $$E_r = \frac{p - \overline{p}}{P} = \frac{3.1415926536 - 3.1428571429}{3.1415926536} = -0.000402$$

www.FreeLibros.me

3. $$E = p - \overline{p} = 2.8182818285 - 2.8182 = -0.099918$$
$$E_r = \frac{p - \overline{p}}{P} = \frac{2.8182818285 - 2.8182}{2.8182818285} = -0.036757$$

4. $$E = p - \overline{p} = 2.8182818285 - 2.7083333333 = -0.009948$$
$$E_r = \frac{p - \overline{p}}{P} = \frac{2.8182818285 - 2.7083333333}{2.8182818285} = -0.036598$$

1.6.6

	Exacta	Truncamiento	Redondeo
$\frac{122}{135} - \frac{11}{32} + \frac{20}{19}$	1.612585283	1.612	1.613
E		0.000585	−0.000414
E_r		0.000362	−0.000257

1.6.7

	Exacta	Truncamiento	Redondeo
$215 - 0.345 - 214$	0.655	0.0000	1.0000
E		0.655	−0.34500
E_r		1.0000	−0.52671
$215 - 214 - 0.345$	0.655	0.655	0.655
E		0	0
E_r		0	0

1.6.8

	Exacta	Truncamiento	Redondeo
x	1	1.25	1.20
y	2	2.50	2.20
$E(x)$		0.25	0.20
$E_r(x)$		0.25	0.20
$E(y)$		0.5	0.20
$E_r(y)$		0.25	0.10

1.6.9

	Exacta	Redondeo
x	3	3.00
y	2	1.99
$E(x)$		0
$E_r(x)$		0.0

www.FreeLibros.me

	Exacta	Redondeo
$E(y)$		0.001
$E_r(y)$		0.005

1.6.10

	Exacta	Truncamiento	Redondeo
$f(x)=\sqrt{x^2+1}-1$	0.00004999875006	0	0
E		0.00004999875006	0.00004999875006
E_r		1.00000000000000	1.00000000000000
$f(x)=\dfrac{x^2}{\sqrt{x^2+1}+1}$	0.00004999875006	0.000050000	0.000050000
E		-0.12499×10^{-8}	-0.12499×10^{-8}
E_r		-0.24999×10^{-4}	-0.24999×10^{-4}

1.6.11 $f(0.01)=0.50167$

	Exacta	Truncamiento	Redondeo
$\dfrac{1}{2!}+\dfrac{1}{3!}x$	0.5016666666	0.501	0.502
E	0.33333×10^{-5}	0.00067	-0.00033
E_r	0.66645×10^{-5}	0.00133	-0.00065

1.6.12 $p(-1.5)=-26.65625$

1.6.13

	Exacta	Truncamiento	Redondeo
$p(x)=x^3+4.12x^2-3.16x+1.34$	0.012243129	0.01700	0.01200
E		-0.475×10^{-2}	0.243×10^{-3}
E_r		-0.388	0.198×10^{-1}

	Exacta	Truncamiento	Redondeo
$p(x)=((x+4.12)x-3.16)x+1.34$	0.012243129	0.01225	0.01225
E		-0.688×10^{-5}	-0.688×10^{-5}
E_r		-0.561×10^{-3}	-0.561×10^{-3}

1.6.14

	Exacta	$r_1, r_2=\dfrac{-b\pm\sqrt{b^2-4ac}}{2a}$	$r_1=signo\left(\dfrac{-b}{2a}\right)\left[\left\|\dfrac{-b}{2a}\right\|+\sqrt{\left(\dfrac{b}{2a}\right)^2-\dfrac{c}{a}}\right];r_2=\dfrac{c/a}{r_1}$
r_1	99999.99999	100000	100000
r_2	0.00001	0	0.00001

www.FreeLibros.me

1.6.15.

	Exacta	$r_1, r_2 = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$	$r_1 = signo\left(\frac{-b}{2a}\right)\left[\left\lvert\frac{-b}{2a}\right\rvert + \sqrt{\left(\frac{b}{2a}\right)^2 - \frac{c}{a}}\right]; r_2 = \frac{c/a}{r_1}$
r_1	-1000	-1000	-1000
r_2	-0.001	0	-0.001

1.6.16

Exacta	$0.6144212353 \times 10^{-5}$
Orden inverso	$0.4700000000 \times 10^{-5}$
Orden normal	-0.9999953194

Capítulo 2

2.11.1 $x_1 = 2.617993$, $x_2 = 2.879793$ y $x_3 = 2.748893$.

2.11.2 $x_1 = 6.5$, $x_2 = 6.25$, $x_3 = 6.375$ y $x_4 = 6.4375$.

2.11.3 $x_1 = 2.5$, $x_2 = 2.25$, $x_3 = 2.125$, $x_4 = 2.0625$ y $x_5 = 2.03125$.

2.11.4 $x_1 = 1.35$, $x_2 = 1.325$, $x_3 = 1.3125$, $x_4 = 1.31875$, $x_5 = 1.315625$ y $x_6 = 1.3140625$.

2.11.5 $x_1 = 5.5$, $x_2 = 5.75$, $x_3 = 5.625$, $x_4 = 5.6875$, $x_5 = 5.71875$, $x_6 = 5.734375$ y $x_7 = 5.7265625$.

2.11.6 Primer cruce: $x_1 = 1.159375$. Segundo cruce: $x_2 = 1.5232421875$.

2.11.7 Primer cruce: $x_1 = 40.5734375$. Segundo cruce: $x_2 = 41.4703125$. Tercer cruce: $x_3 = 45.669921875$. Cuarto cruce: $x_4 = 49.11484375$. Quinto cruce: $x_5 = 51.07265625$.

2.11.8 $x_1 = 1.442957$, $x_2 = 1.509555$, $x_3 = 1.518042$ y $x_4 = 1.519100$.

2.11.9 $x_1 = 36.280587$, $x_2 = 36.266435$, $x_3 = 36.266644$ y $x_4 = 36.266644$.

2.11.10 $x_1 = 2.760192$, $x_2 = 2.855380$ y $x_3 = 2.852299$.

2.11.11 $x_1 = 1.344265$, $x_2 = 1.079525$, $x_3 = 1.019967$, $x_4 = 1.061810$ y $x_5 = 1.061076$.

2.11.12 Primer cruce: $x_1 = 1.835539$. Segundo cruce: $x_2 = 2.509761$. Tercer cruce: $x_3 = 3.180363$.

2.11.13 Primer cruce: $x_1 = -0.070775$. Segundo cruce: $x_2 = 0.812500$. Tercer cruce: $x_3 = 1.856820$.

2.11.14 $x_2 = 5$.

2.11.15 $x_2 = 1.028004$, $x_3 = 1.029753$, $x_4 = 1.029866$ y $x_5 = 1.029866$.

2.11.16 $x_2 = 4.186091$, $x_3 = 4.152591$, $x_4 = 4.150614$ y $x_5 = 4.150683$.

2.11.17 $x_2 = 0.014465366$ y $x_3 = 0.014465257$.

2.11.18 $x_2 = 1.500191$, $x_3 = 1.464145$, $x_4 = 1.462237$ y $x_5 = 1.462375$.

2.11.19 $x_2 = 1.981043$, $x_3 = 1.760559$, $x_4 = 1.793251$, $x_5 = 1.795496$ y $x_6 = 1.795465$.

2.11.20 $x_2 = -0.508904$, $x_3 = 0.298669$, $x_4 = 0.297344$, $x_5 = -0.203159$, $x_6 = 0.279919$, $x_7 = 0.263723$, $x_8 = -0.091448$, $x_9 = 0.228353$, $x_{10} = 0.200250$, $x_{11} = 0.053270$, $x_{12} = 0.143315$, $x_{13} = 0.124966$, $x_{14} = 0.116309$, $x_{15} = 0.117383$ y $x_{16} = 0.117344$.

2.11.21 $x_1 = 0.447981$, $x_2 = 0.137751$, $x_3 = 0.126187$, $x_4 = 0.125996$ y $x_5 = 0.125993$.

2.11.22 $x_1 = 1.368988$, $x_2 = 1.356029$, $x_3 = 1.358289$ y $x_4 = 1.357909$.

2.11.23 Con $x_0 = 1$: $x_1 = 13.903221$, $x_2 = 20.081980$ y $x_3 = 20.085521$. Con $x_0 = 50$: $x_1 = 20.085536$ y $x_2 = 20.085521$.

www.FreeLibros.me

2.11.24 $x_1 = 24.980915$ y $x_2 = 24.980915$.

2.11.25 $x_1 = 34.231587$, $x_2 = 34.228289$ y $x_3 = 34.228335$.

2.11.26 $x_1 = 24.856681$, $x_2 = 24.384042$, $x_3 = 24.520598$, $x_4 = 24.530702$ y $x_5 = 24.530764$.

2.11.27 $x_1 = 1.333333$, $x_2 = 1.408579$, $x_3 = 1.412381$ y $x_4 = 1.412391$.

2.11.28 $x_1 = 0.321114$, $x_2 = 0.316753$ y $x_3 = 0.316750$.

2.11.29 $x_0 = 16.4$, $x_1 = 16.340652$, $x_2 = 16.357129$ y $x_3 = 16.357494$.

2.11.30 $x_0 = 7$, $x_1 = 7.890293$, $x_2 = 7.954565$, $x_3 = 7.971583$, $x_4 = 7.976124$, $x_3 = 16.357494$, $x_5 = 7.977337$, $x_6 = 7.977661$, $x_7 = 7.977747$, $x_8 = 7.977770$, $x_9 = 7.977776$, $x_{10} = 7.977778$ y $x_{11} = 7.977779$.

2.11.31

x	1	7.254646	4.582485	3.597040	3.254273	3.217014	3.217066	3.217066
y	1	−1.729959	−0.223884	0.429718	0.512009	0.520032	0.520951	0.520950
z	1	4.200810	2.989813	2.766001	2.781328	2.784352	2.784257	2.784257

2.11.32

x	y	z
10	10	10
12.7851280105231	3.9500746224528	6.85403914292963
−45.2366712835505	7.39804635655532	14.5477648099304
−7.52457266042121	5.59208036597501	15.1805431754463
−4.72318885384847	3.7620556320533	13.1111945173895
−2.96048308698226	2.6029816470485	11.3433620711519
−1.88075744591476	1.81398372421956	9.96263306073305
−1.23618140857338	1.25954974237131	9.04791407252535
−0.875438909842467	0.881411960260863	8.70795059456642
−0.713844401005653	0.675259405162648	8.96387338456751
−0.68128457226444	0.627516583056792	9.28061711691572
−0.681413283695717	0.627047863522449	9.30580796916884
−0.6814159609685	0.627048062274151	9.30577024476398
−0.681415961008203	0.627048062289772	9.30577024451167

2.11.33

x	y	z
1	1	1
−11.9637526652452	11.2878464818763	3.2089552238806
−9.10720240434747	8.33075063911712	4.11836873804045
−6.90313724210736	6.17265545296276	3.38054392058566
−5.10605230340714	4.66480010842117	2.39012595511254
−3.62906806358636	3.60497070390569	1.20213531006073
−2.54958605402661	2.69087610294776	−0.705211634556044
−2.28995657281888	1.91846137406384	−5.65400962686616
−1.94710734616362	2.5501292258611	−4.21721575654591

www.FreeLibros.me

x	*y*	*z*
−1.38067133408591	2.53206000243663	−3.59444157682631
−1.05384756763737	2.53138442158787	−3.32598024143683
−0.941033728609823	2.52727726549691	−3.26783821123274
−0.929305879384688	2.52606355063758	−3.26522284185036
−0.929205299113197	2.52604523140529	−3.26523125827778
−0.929205293431239	2.52604522970842	−3.26523126133965

x	*y*	*z*
1	2	3
−0.536483873462758	7.62275504679403	0.566865186789389
−0.378219012651542	5.71815979640691	3.43759513928358
−0.192862692275438	4.46870756399213	2.59666391926414
0.0821970487682144	4.16816652531562	2.2291612476594
0.141247103043527	4.20319813941261	2.30307702456874
0.138715886849022	4.20401410277074	2.30509356889384
0.138716391835808	4.20401754172374	2.30510038605867
0.138716391834323	4.20401754171813	2.30510038604467

2.11.34 Un sistema no lineal puede tener varias soluciones. Dependiendo de las condiciones inícia-les, puede converger a la más cercana o a la más dominante.

2.11.35

Gauss			Gauss-Seidel		
4	7	3	4	7	3
−2.285714	2.034482	8.153846	−2.285714	2.406052	1.483323
−0.566448	1.949727	0.958492	3.366607	0.392165	3.567622
3.038635	0.596052	2.909072	1.201319	0.508763	3.309114
1.507503	0.274778	3.592277	1.450336	0.515420	3.349001
1.316666	0.437540	3.327354	1.408066	0.523992	3.342968
1.444926	0.504618	3.328648	−2.285714	2.406052	1.483323
1.422324	0.515561	3.346747	1.412901	0.526164	3.343946
1.411387	0.522529	3.344187	1.411542	0.527101	3.343806
1.412251	0.525872	3.343464	1.411535	0.527418	3.343832
1.411925	0.526905	3.343827	1.411456	0.527537	3.343830
1.411534	0.527314	3.343844	1.411440	0.527580	3.343831
1.411471	0.527494	3.343823	1.411432	0.527595	3.343831
1.411451	0.527561	3.343829	1.411429	0.527601	3.343831
1.411436	0.527587	3.343831	1.411428	0.527603	3.343831
1.411431	0.527597	3.343831	1.411428	0.527604	3.343831
1.411429	0.527601	3.343831			
1.411428	0.527603	3.343831			
1.411428	0.527604	3.343831			

www.FreeLibros.me

Gauss			Gauss–Seidel		
4	7	8	4	7	8
−5.142857	5.655172	5.461538	−5.142857	5.294862	−6.234697
4.462512	2.696856	−5.779782	10.881359	−7.084123	−17.878618
5.012053	−1.885370	7.283963	24.657052	−7.448524	−49.318869
0.616246	−1.873089	2.183402	57.847733	−76.975926	−1316.308785
2.345810	−0.022375	3.355911	1391.731143	−56307.976232	−2.378579e+7
1.518405	0.216272	3.301355	2.478696e+7	−2.104743e+13	−1.589028e+20
1.495169	0.408152	3.328551	1.653304e+20	−9.425572e+38	−4.748272e+58
1.439362	0.477246	3.344016	4.939490e+58	−8.413299e+115	−1.266314e+174
1.419571	0.507448	3.343451	1.317285e+174	−Inf	−Inf
1.415119	0.519817	3.343419			
1.412960	0.524528	3.343757			
1.411976	0.526387	3.343809			
1.411645	0.527128	3.343815			
1.411517	0.527418	3.343825			
1.411462	0.527531	3.343829			
1.411441	0.527575	3.343830			
1.411433	0.527593	3.343831			
1.411430	0.527600	3.343831			
1.411429	0.527602	3.343831			
1.411428	0.527603	3.343831			
1.411428	0.527604	3.343831			

Capítulo 3

3.8.1 $r_1=-4,\ r_2=5,\ r_3=4,\ r_4=-1,\ r_5=3,\ r_6=2$ y $r_7=1$.

3.8.2 $x^5-17x^4+59x^3+233x^2-1\,140x+864$.

3.8.3 $z^6-7z^5+7z^4+47z^3-140z^2+140z-48$.

3.8.4 $y^6-3y^5-65y^4+135y^3+1\,144y^2-1\,212y-5\,040$.

3.8.5 a^2-1.

3.8.6 $d^4+4d^3-23d^2-54d+72$.

3.8.7 $y^3-6y^2-19y+84$.

3.8.8 $y^4-19y^3+113y^2-221y+126$.

3.8.9 $y^4+11y^3+41y^2+61y+30$.

3.8.10 $y^3+27y^2+179y+153$.

3.8.11 $y^6+26y^5+250y^4+1\,160y^3+2\,749y^2+3\,134y+1\,320$.

3.8.12 $P'(z)=105$.

3.8.13 $P_1'(z)=-15\,444,\ P_2'(z)=3\,528,\ P_3'(z)=-3\,348,\ P_4'(z)=12\,740$ y $P_5'(z)=668\,360$.

3.8.14 Original: $r_1=-9,\ r_2=-4,\ r_3=5,\ r_4=3$ y $r_5=2$. Inverso: $r_1=-0.\overline{1}1,\ r_2=-0.25,\ r_3=0.2,$ $r_4=0.\overline{33}$ y $r_5=0.5$.

www.FreeLibros.me

3.8.15 x^2+3x+2.

3.8.16 $f_1(x)=x^2+2x+1,\ f_2(x)=x^2+12x+36$ y $f_3(x)=x^2+20x+99$.

3.8.17 $f_1(x)=x^2+11x+18$ y $f_2(x)=x^2+12x+32$.

3.8.18 $f_1(x)=x^2+1,\ f_2(x)=x^2-8x+29,\ f_3(x)=x^2+5x+36$ y $f_4(x)=x+6$.

3.8.19 $f_1(x)=x^2+0.689141x+0.180778,\ f_2(x)=x^2-0.537824x+0.228049,$
$f_3(x)=x^2+3.668790x+5.545771,\ f_4(x)=x^2-2.038036x+2.497077$ y
$f_5(x)=x^2+7.217929x+40.286468$.

3.8.20 $r=4$.

3.8.21 $r=-4$.

3.8.22 $r=-24$.

3.8.23 $r=-1$.

3.8.24 $r_1=-4,\ r_2=-7,\ r_3=-8,\ r_4=-9,\ r_5=-10,\ r_6=-11,\ r_7=-12,\ r_8=-13,\ r_9=-14$ y $r_{10}=-15$.

3.8.25 $r=-13$.

3.8.26 $u_1=-27,\ u_2=276.999999,\ u_3=-3320.999999$ y $u_4=42768.999999$.

3.8.27 $r_1=-14,\ r_2=-10,\ r_3=-8,\ r_4=-6,\ r_5=-4$ y $r_6=-3$.

3.8.28 $r_1=-15,\ r_2=-13,\ r_3=-11,\ r_4=-9,\ r_5=-7,\ r_6=-5,\ r_7=-3$ y $r_8=-1$.

3.8.29 $r_1=-12,\ r_2=-10,\ r_3=-8,\ r_4=-6,\ r_5=-4$ y $r_6=-2$.

3.8.30 $r=-0.499999$.

3.8.31 $r_1=-0.2,\ r_2=-0.499999,\ r_3=-0.7,\ r_4=-1.199999,\ r_5=-2.3,\ r_6=-3.399999$ y $r_7=-8$.

3.8.32 $r_1=-0.168488+0.917716i,\ r_2=-0.168488-0.917716i,\ r_3=-4.524183,\ r_4=1.430580-5.609533i$
y $r_5=1.430580+5.609533i$.

3.8.33 $r_1=-0.040154,\ r_2=-1.574491+2.124589i,\ r_3=-1.574491-2.124589i,\ r_4=-4.860059+2.680353i,$
$r_5=-4.860059-2.680353i,\ r_6=-7.545371+1.155189i$ y $r_7=-7.545371-1.155189i$.

3.8.34 $r_1=-0.01,\ r_2=-0.2,\ r_3=-1,\ r_4=-0.999999,\ r_5=-2,\ r_6=-1.999999$ y $r_7=-3$.

3.8.35

	q^1	ε^1	q^2	ε^2	q^3	ε^3	q^4	
	−29.0000		0		0		0	
0		9.9310		3.8750		1.1613		0
	−19.0690		−6.0560		−2.7137		−1.1613	
0		3.1540		1.7364		0.4970		0
	−15.9150		−7.4736		−3.9531		−1.6582	
0		1.4811		0.9185		0.2085		0
	−14.4339		−8.0362		−4.6631		−1.8667	
0		0.8246		0.5329		0.0834		0
	−13.6093		−8.3279		−5.1126		−1.9502	
0		0.5046		0.3272		0.0318		0
	−13.1047		−8.5053		−5.4080		−1.9820	
0		0.3275		0.2080		0.0117		0
	−12.7772		−8.6248		−5.6044		−1.9937	

www.FreeLibros.me

	q^1	ε^1	q^2	ε^2	q^3	ε^3	q^4	
0		0.2211		0.1352		0.0041		0
	−12.5562		−8.7106		−5.7354		−1.9978	
0		0.1534		0.0890		0.0014		0
	−12.4028		−8.7750		−5.8230		−1.9992	
0		0.1085		0.0591		0.0005		0
	−12.2943		−8.8244		−5.8815		−1.9997	
0		0.0779		0.0394		0.0002		0
	−12.2164		−8.8629		−5.9207		−1.9999	
0		0.0565		0.0263		0.0001		0
	−12.1599		−8.8931		−5.9470		−2.0000	
0		0.0413		0.0176		0.0000		0
	−12.1186		−8.9169		−5.9645		−2.0000	
0		0.0304		0.0118		0.0000		0
	−12.0882		−8.9355		−5.9763		−2.0000	

3.8.36

	q^1	ε^1	q^2	ε^2	q^3	ε^3	q^4	
	−17.0000		0		0		0	
0		5.9412		2.4455		0.8502		0
	−11.0588		−3.4956		−1.5953		−0.8502	
0		1.8780		1.1161		0.4531		0
	−9.1809		−4.2575		−2.2583		−1.3033	
0		0.8709		0.5920		0.2615		0
	−8.3100		−4.5364		−2.5889		−1.5648	
0		0.4754		0.3379		0.1580		0
	−7.8346		−4.6739		−2.7687		−1.7228	
0		0.2836		0.2001		0.0983		0
	−7.5509		−4.7574		−2.8705		−1.8212	
0		0.1787		0.1208		0.0624		0
	−7.3722		−4.8153		−2.9289		−1.8836	
0		0.1167		0.0735		0.0401		0
	−7.2555		−4.8586		−2.9622		−1.9237	
0		0.0782		0.0448		0.0261		0
	−7.1774		−4.8920		−2.9809		−1.9498	
0		0.0533		0.0273		0.0170		0
	−7.1241		−4.9179		−2.9912		−1.9668	
0		0.0368		0.0166		0.0112		0
	−7.0873		−4.9381		−2.9966		−1.9780	
0		0.0256		0.0101		0.0074		0
	−7.0617		−4.9537		−2.9992		−1.9854	

www.FreeLibros.me

	q^1	ε^1	q^2	ε^2	q^3	ε^3	q^4	
0		0.0180		0.0061		0.0049		0
	−7.0437		−4.9655		−3.0004		−1.9903	
0		0.0127		0.0037		0.0032		0
	−7.0311		−4.9745		−3.0009		−1.9935	
0		0.0090		0.0022		0.0022		0
	−7.0221		−4.9813		−3.0009		−1.9957	
0		0.0064		0.0013		0.0014		0
	−7.0157		−4.9863		−3.0008		−1.9971	
0		0.0045		0.0008		0.0010		0
	−7.0112		−4.9900		−3.0007		−1.9981	

3.8.37

	q^1	ε^1	q^2	ε^2	q^3	ε^3	q^4	ε^4	q^5	
	17.0000		0		0		0		0	
0		−6.2941		−2.8692		−1.2899		−0.4545		0
	10.7059		3.4250		1.5793		0.8354		0.4545	
0		−2.0136		−1.3230		−0.6823		−0.2473		0
	8.6923		4.1156		2.2199		1.2703		0.7019	
0		−0.9534		−0.7136		−0.3904		−0.1367		0
	7.7389		4.3553		2.5431		1.5241		0.8385	
0		−0.5365		−0.4167		−0.2340		−0.0752		0
	7.2024		4.4752		2.7258		1.6829		0.9137	
0		−0.3334		−0.2538		−0.1445		−0.0408		0
	6.8690		4.5547		2.8351		1.7866		0.9545	
0		−0.2211		−0.1580		−0.0910		−0.0218		0
	6.6480		4.6178		2.9021		1.8558		0.9764	
0		−0.1535		−0.0993		−0.0582		−0.0115		0
	6.4944		4.6721		2.9431		1.9025		0.9878	
0		−0.1105		−0.0625		−0.0376		−0.0060		0
	6.3840		4.7200		2.9681		1.9342		0.9938	
0		−0.0817		−0.0393		−0.0245		−0.0031		0
	6.3023		4.7623		2.9829		1.9557		0.9968	
0		−0.0617		−0.0246		−0.0161		−0.0016		0
	6.2406		4.7994		2.9914		1.9702		0.9984	
0		−0.0475		−0.0154		−0.0106		−0.0008		0
	6.1931		4.8315		2.9962		1.9800		0.9992	
0		−0.0370		−0.0095		−0.0070		−0.0004		0
	6.1561		4.8590		2.9987		1.9866		0.9996	
0		−0.0292		−0.0059		−0.0046		−0.0002		0
	6.1269		4.8824		2.9999		1.9910		0.9998	

www.FreeLibros.me

	q^1	ε^1	q^2	ε^2	q^3	ε^3	q^4	ε^4	q^5	
0		−0.0233		−0.0036		−0.0031		−0.0001		0
	6.1036		4.9021		3.0005		1.9940		0.9999	
0		−0.0187		−0.0022		−0.0020		−0.0001		0
	6.0849		4.9186		3.0006		1.9960		0.9999	

3.8.38

	q^1	ε^1	q^2	ε^2	q^3	ε^3	q^4	ε^4	q^5	ε^5	q^6	
	−59.00		0		0		0		0		0	
0		22.54		10.74		5.18		2.22		0.62		0
	−36.45		−11.79		−5.55		−2.96		−1.59		−0.62	
0		7.29		5.06		2.76		1.19		0.24		0
	−29.16		−14.03		−7.85		−4.53		−2.54		−0.87	
0		3.51		2.83		1.59		0.67		0.08		0
	−25.65		−14.71		−9.08		−5.46		−3.12		−0.96	
0		2.01		1.74		0.96		0.38		0.02		0
	−23.63		−14.97		−9.87		−6.03		−3.48		−0.98	
0		1.27		1.15		0.58		0.22		0.00		0
	−22.36		−15.09		−10.43		−6.40		−3.70		−0.99	
0		0.86		0.79		0.36		0.12		0.00		0
	−21.50		−15.16		−10.87		−6.63		−3.82		−0.99	
0		0.60		0.57		0.21		0.07		0.00		0
	−20.89		−15.19		−11.22		−6.77		−3.90		−0.99	
0		0.44		0.42		0.13		0.04		0.00		0
	−20.45		−15.21		−11.51		−6.86		−3.94		−1.00	
0		0.32		0.31		0.07		0.02		0.00		0
	−20.12		−15.22		−11.75		−6.92		−3.96		−1.00	
0		0.24		0.24		0.04		0.01		0.00		0
	−19.87		−15.22		−11.95		−6.95		−3.98		−1.00	
0		0.19		0.19		0.02		0.00		0.00		0
	−19.68		−15.22		−12.12		−6.97		−3.98		−1.00	
0		0.14		0.15		0.01		0.00		0.00		0
	−19.54		−15.21		−12.26		−6.98		−3.99		−1.00	
0		0.11		0.12		0.00		0.00		0.00		0
	−19.42		−15.20		−12.37		−6.99		−3.99		−1.00	
0		0.08		0.10		0.00		0.00		0.00		0
	−19.33		−15.19		−12.47		−6.99		−3.99		−1.00	
0		0.07		0.08		0.00		0.00		0.00		0
	−19.26		−15.18		−12.55		−6.99		−3.99		−1.00	
0		0.05		0.06		0.00		0.00		0.00		0
	−19.21		−15.16		−12.62		−6.99		−3.99		−1.00	

www.FreeLibros.me

	q^1	ε^1	q^2	ε^2	q^3	ε^3	q^4	ε^4	q^5	ε^5	q^6	
0		0.04		0.05		0.00		0.00		0.00		0
	−19.16		−15.15		−12.67		−6.99		−3.99		−1.00	
0		0.03		0.04		0.00		0.00		0.00		0
	−19.13		−15.14		−12.72		−6.99		−3.99		−1.00	
0		0.02		0.04		0.00		0.00		0.00		0
	−19.10		−15.13		−12.76		−6.99		−3.99		−1.00	
0		0.02		0.03		0.00		0.00		0.00		0
	−19.08		−15.11		−12.79		−6.99		−3.99		−1.00	
0		0.01		0.02		0.00		0.00		0.00		0
	−19.06		−15.10		−12.82		−6.99		−4.00		−1.00	
0		0.01		0.02		0.00		0.00		0.00		0
	−19.05		−15.09		−12.85		−7.00		−4.00		−1.00	
0		0.01		0.02		0.00		0.00		0.00		0
	−19.04		−15.08		−12.87		−7.00		−4.00		−1.00	
0		0.00		0.01		0.00		0.00		0.00		0
	−19.03		−15.07		−12.89		−7.00		−4.00		−1.00	
0		0.00		0.01		0.00		0.00		0.00		0
	−19.02		−15.06		−12.90		−7.00		−4.00		−1.00	
0		0.00		0.01		0.00		0.00		0.00		0
	−19.02		−15.06		−12.91		−7.00		−4.00		−1.00	

3.8.39

	q^1	ε^1	q^2	ε^2	q^3	ε^3	q^4	ε^4	q^5	ε^5	q^6	ε^6	q^7	ε^7	q^8	
	−43.0		0		0		0		0		0		0		0	
0		18.2		10.0		6.0		3.6		2.1		1.1		0.4		0
	−24.7		−8.1		−4.0		−2.3		−1.5		−1.0		−0.6		−0.4	
0		6.0		4.9		3.5		2.3		1.4		0.7		0.2		0
	−18.7		−9.2		−5.4		−3.5		−2.4		−1.7		−1.1		−0.6	
0		2.9		2.9		2.2		1.6		1.0		0.5		0.1		0
	−15.7		−9.3		−6.0		−4.1		−3.0		−2.2		−1.5		−0.8	
0		1.7		1.8		1.5		1.1		0.7		0.3		0.0		0
	−14.0		−9.1		−6.3		−4.6		−3.4		−2.5		−1.8		−0.9	
0		1.1		1.3		1.1		0.8		0.5		0.2		0.0		0
	−12.8		−8.9		−6.5		−4.9		−3.7		−2.8		−2.1		−0.9	
0		0.8		0.9		0.8		0.6		0.4		0.2		0.0		0
	−12.0		−8.8		−6.6		−5.1		−3.9		−3.0		−2.3		−0.9	
0		0.5		0.7		0.6		0.5		0.3		0.1		0.0		0
	−11.4		−8.6		−6.7		−5.2		−4.1		−3.2		−2.4		−0.9	
0		0.4		0.5		0.5		0.4		0.2		0.1		0.0		0
	−11.0		−8.5		−6.7		−5.3		−4.2		−3.3		−2.5		−0.9	

www.FreeLibros.me

	q^1	ε^1	q^2	ε^2	q^3	ε^3	q^4	ε^4	q^5	ε^5	q^6	ε^6	q^7	ε^7	q^8	
0		0.3		0.4		0.4		0.3		0.2		0.0		0.0		0
	−10.7		−8.4		−6.7		−5.4		−4.3		−3.4		−2.6		−0.9	
0		0.2		0.3		0.3		0.2		0.1		0.0		0.0		0
	−10.4		−8.3		−6.8		−5.5		−4.4		−3.5		−2.7		−0.9	
0		0.2		0.2		0.2		0.2		0.1		0.0		0.0		0
	−10.2		−8.3		−6.8		−5.5		−4.5		−3.6		−2.8		−0.9	
0		0.1		0.2		0.2		0.1		0.1		0.0		0.0		0
	−10.0		−8.2		−6.8		−5.6		−4.6		−3.7		−2.8		−1.0	
0		0.1		0.1		0.1		0.1		0.0		0.0		0.0		0
	−9.8		−8.2		−6.8		−5.6		−4.6		−3.7		−2.8		−1.0	
0		0.1		0.1		0.1		0.1		0.0		0.0		0.0		0
	−9.7		−8.1		−6.8		−5.7		−4.7		−3.8		−2.9		−1.0	
0		0.1		0.1		0.1		0.0		0.0		0.0		0.0		0
	−9.6		−8.1		−6.8		−5.7		−4.7		−3.8		−2.9		−1.0	
0		0.0		0.1		0.1		0.0		0.0		0.0		0.0		0
	−9.5		−8.1		−6.8		−5.7		−4.7		−3.8		−2.9		−1.0	

3.8.40

0	16.155	13.004	12.149	12.007	12	12
0	8.702	8.799	8.936	8.995	9	9
0	4.609	5.663	5.969	5.999	6	

3.8.41

0	7.681	7.059	7.001	7
0	2.908	2.984	3	3
0	0.94	0.997	1	1

3.8.42

0	17.72	13.38	12.2	12.01	12	12
0	10.3	9.56	9.27	9.07	9.01	9
0	6.48	7.18	7.68	7.93	7.99	8
0	3.65	4.7	4.98	5	5	5

3.8.43

0	16.64	12.48	11.29	11.03	11
0	9.73	9.04	8.95	8.99	9
0	6.17	6.68	6.93	6.99	7
0	3.6	4.6	4.9	5	5
0	0.96	1	1	1	1

3.8.44

0	18.63	14.28	13.2	13.01	13
0	10.66	9.93	9.93	9.99	10

www.FreeLibros.me

0	6.63	6.92	7	7	7
0	3.95	4.67	4.96	5	5
0	1.75	1.98	2	2	2

3.8.45

0	25.4	18.97	17.3	17.02	17
0	14.91	13.55	13.21	13.04	13
0	9.63	10.12	10.69	10.95	11
0	6.01	6.76	6.98	7	7
0	3.26	3.88	3.99	4	4
0	0.95	1	1	1	1

3.8.46 $r_1 = 0.144008 + 0.855235i$, $r_2 = 0.119858 - 0.916136i$, $r_3 = -1.024852 - 0.494128i$, $r_4 = 1.061858 + 0.988144i$ y $r_5 = -12.771460 - 6.550761i$.

3.8.47 $r_1 = 0.999999 + 0.999999i$, $r_2 = 4 + 2.999999i$, $r_3 = 1 + 1i$, $r_4 = 3.998999 + 3i$ y $r_5 = 4 - 3i$.

3.8.48 $r_1 = 0.088309 - 0.780864i$, $r_2 = 0.456098 - 0.051378i$, $r_3 = -0.516261 + 0.702056i$, $r_4 = -0.831892 - 0.128205i$, $r_5 = 0.548658 + 1.120551i$ y $r_5 = 0.475270 - 1.128215i$.

3.8.49 $r_1 = -0.347039 + 0.230086i$, $r_2 = 0.497926 + 0.290230i$, $r_3 = -0.218588 - 0.747737i$, $r_4 = -1.595043 - 0.673868i$, $r_5 = 1.390044 - 0.363263i$ y $r_5 = -2.100491 + 2.571327i$.

Capítulo 4

4.7.1 $x_1 = -3$, $x_2 = 2.555555$ y $x_3 = 0.888888$.

4.7.2 $x_1 = 0.393468$, $x_2 = 0.178849$, $x_3 = 0.625194$ y $x_4 = -0.307931$.

4.7.3 $x_1 = -0.799796$, $x_2 = 0.552845$, $x_3 = -0.286585$ y $x_4 = 1.043699$.

4.7.4 $x_1 = 1.320895$, $x_2 = -1.399253$ y $x_3 = 0.309701$.

4.7.5 $x_1 = -1.141987$, $x_2 = 1.898580$, $x_3 = 0.876267$ y $x_4 = 0.537525$.

4.7.6 $x_1 = 1.064516$, $x_2 = 1.241935$, $x_3 = 0.370967$ y $x_4 = -4.338709$.

4.7.7 $x_1 = 0.375$, $x_2 = -0.125$ y $x_3 = 0$.

4.7.8
$$\mathbf{A} = \begin{bmatrix} 0.4767 & 0.1819 & -0.1820 & -0.0792 & -0.2621 & 0.0669 \\ 0.5966 & 0.1862 & -0.0859 & 0.0084 & -0.3259 & -0.1338 \\ -0.0087 & 0.1614 & -0.0325 & 0.0447 & -0.0209 & -0.0578 \\ 0.0219 & -0.1141 & 0.1410 & -0.2308 & 0.1117 & 0.1463 \\ -0.4796 & -0.3503 & 0.2823 & -0.0170 & 0.2551 & 0.1361 \\ -0.5566 & -0.0736 & -0.1043 & 0.2430 & 0.3046 & -0.0442 \end{bmatrix}$$

4.7.9
$$\mathbf{A} = \begin{bmatrix} -0.0984 & 0.3443 & -0.0984 & -0.0820 \\ -0.0179 & -0.2101 & 0.0730 & 0.1669 \\ -0.0849 & 0.2519 & 0.0969 & -0.2072 \\ 0.3681 & -0.4247 & -0.0864 & 0.2310 \end{bmatrix}$$

4.7.10
$$\mathbf{A} = \begin{bmatrix} -0.0526 & 0.2105 & 0.2632 \\ 0.2105 & -0.8421 & -0.0526 \\ -0.2105 & 1.8421 & 0.0526 \end{bmatrix}$$

www.FreeLibros.me

4.7.11 $\mathbf{L}=\begin{bmatrix}1&0&0&0\\3&1&0&0\\6&8&1&0\\7&5&3&1\end{bmatrix},\ \mathbf{U}=\begin{bmatrix}4&6&8&1\\0&3&6&8\\0&0&3&7\\0&0&0&3\end{bmatrix}$

4.7.12 $\mathbf{L}=\begin{bmatrix}1&0&0&0\\2&1&0&0\\3&1&1&0\\2&7&5&1\end{bmatrix},\ \mathbf{U}=\begin{bmatrix}9&5&2&9\\0&7&4&2\\0&0&2&1\\0&0&0&1\end{bmatrix}$

4.7.13 $\mathbf{L}=\begin{bmatrix}1&0&0&0\\7&1&0&0\\1&0&1&0\\2&5&9&1\end{bmatrix},\ \mathbf{U}=\begin{bmatrix}3&8&2&2\\0&2&1&7\\0&0&2&1\\0&0&0&1\end{bmatrix}$

4.7.14 $\mathbf{L}=\begin{bmatrix}1&0&0&0\\0.1&1&0&0\\1.8&1.6&1&0\\2.4&2.5&0.9&1\end{bmatrix},\ \mathbf{U}=\begin{bmatrix}0.6&1.0&2.2&4.2\\0&0.9&3.1&1.7\\0&0&1.2&1.3\\0&0&0&1.8\end{bmatrix}$

4.7.15 $\mathbf{D}=\begin{bmatrix}4&0&0&0\\5&2&0&0\\8&1&1&0\\7&3&2&9\end{bmatrix},\ \mathbf{C}=\begin{bmatrix}1&4&7&9\\0&1&2&3\\0&0&1&5\\0&0&0&1\end{bmatrix}$

4.7.16 $\mathbf{D}=\begin{bmatrix}7&0&0\\1&9&0\\2&1&4\end{bmatrix},\ \mathbf{C}=\begin{bmatrix}1&2&5\\0&1&7\\0&0&1\end{bmatrix}$

4.7.17 $\mathbf{D}=\begin{bmatrix}7&0&0&0\\3&8&0&0\\9&3&1&0\\1&3&5&7\end{bmatrix},\ \mathbf{C}=\begin{bmatrix}1&2&3&1\\0&1&9&6\\0&0&1&3\\0&0&0&1\end{bmatrix}$

4.7.18 $\mathbf{D}=\begin{bmatrix}3.4&0&0&0\\2.3&2.4&0&0\\3.2&1.4&4&0\\1.2&3.5&4.2&6\end{bmatrix},\ \mathbf{C}=\begin{bmatrix}1&4.4&3.2&1.8\\0&1&2.2&2.7\\0&0&1&1.6\\0&0&0&1\end{bmatrix}$

4.7.19 $\mathbf{C}_{Inf}=\begin{bmatrix}3&0&0&0\\6&2&0&0\\3&1&1&0\\2&3&2&5\end{bmatrix},\ \mathbf{C}_{Sup}=\begin{bmatrix}3&6&3&2\\0&2&1&3\\0&0&1&2\\0&0&0&5\end{bmatrix}$

www.FreeLibros.me

4.7.20 $$\mathbf{C}_{Inf}=\begin{bmatrix}5.3852 & 0 & 0 & 0 & 0\\0.9285 & 7.7549 & 0 & 0 & 0\\1.4856 & 0.8537 & 9.4902 & 0 & 0\\1.6713 & 1.3473 & 0.2494 & 7.7672 & 0\\0.1857 & 0.4936 & 3.2984 & 0.0260 & 3.5835\end{bmatrix},$$

$$\mathbf{C}_{Sup}=\begin{bmatrix}5.3852 & 0.9285 & 1.4856 & 1.6713 & 0.1857\\0 & 7.7549 & 0.8537 & 1.3473 & 0.4936\\0 & 0 & 9.4902 & 0.2494 & 3.2984\\0 & 0 & 0 & 7.7672 & 0.0260\\0 & 0 & 0 & 0 & 3.5835\end{bmatrix}$$

4.7.21 $$\mathbf{C}_{Inf}=\begin{bmatrix}5.2915 & 0 & 0 & 0 & 0\\0.7559 & 5.6061 & 0 & 0 & 0\\0.1890 & 0.3313 & 8.6519 & 0 & 0\\1.3229 & 0.7135 & 0.9840 & 7.8596 & 0\\0.1890 & 0.1529 & 0.9147 & 0.2215 & 4.2491\end{bmatrix},$$

$$\mathbf{C}_{Sup}=\begin{bmatrix}5.2915 & 0.7559 & 0.1890 & 1.3229 & 0.1890\\ & 5.6061 & 0.3313 & 0.7135 & 0.1529\\ & & 8.6519 & 0.9840 & 0.9147\\ & & & 7.8596 & 0.2215\\ & & & & 4.2491\end{bmatrix}$$

4.7.22 $$\mathbf{C}_{Inf}=\begin{bmatrix}9.4340 & 0 & 0 & 0 & 0 & 0\\1.1660 & 7.5921 & 0 & 0 & 0 & 0\\0.8480 & 1.4503 & 8.8982 & 0 & 0 & 0\\2.3320 & 0.5639 & 1.8211 & 8.0578 & 0 & 0\\1.3780 & 2.8178 & -0.3658 & 0.4795 & 9.4232 & 0\\0.7420 & 0.0178 & 2.3988 & -0.2617 & 4.9803 & 6.9154\end{bmatrix},$$

$$\mathbf{C}_{Sup}=\begin{bmatrix}9.4340 & 1.1660 & 0.8480 & 2.3320 & 1.3780 & 0.7420\\0 & 7.5921 & 1.4503 & 0.5639 & 2.8178 & 0.0178\\0 & 0 & 8.8982 & 1.8211 & -0.3658 & 2.3988\\0 & 0 & 0 & 8.0578 & 0.4795 & -0.2617\\0 & 0 & 0 & 0 & 9.4232 & 4.9803\\0 & 0 & 0 & 0 & 0 & 6.9154\end{bmatrix}$$

4.7.23 $x_1=0.450704$, $x_2=0.633802$, $x_3=-0.507042$, $x_4=0.802816$ y $x_5=-2.112676$.

4.7.24 $x_1=-1.130841$, $x_2=0.990654$, $x_3=1.644859$ y $x_4=0.897196$.

4.7.25 $x_1=6.125$, $x_2=13.375$, $x_3=-8.625$ y $x_4=-12.5$.

4.7.26 $x_1=-0.114942$, $x_2=-0.249042$, $x_3=0.432950$ y $x_4=0.812260$.

www.FreeLibros.me

4.7.27 Jacobi: Radio espectral – 0.7919, número de iteraciones – 30

x_1	0	0.2500	0.3333	0.3958	0.3889	0.4010	0.3877	0.3940	0.3864	0.3912
x_2	0	0.2500	0.2708	0.2292	0.1962	0.1753	0.1685	0.1654	0.1660	0.1658
x_3	0	0.3333	0.2500	0.1806	0.0903	0.0752	0.0492	0.0580	0.0496	0.0576

x_1	0.3870	0.3903	0.3877	0.3898	0.3882	0.3895	0.3884	0.3892	0.3886
x_2	0.1666	0.1664	0.1668	0.1666	0.1668	0.1666	0.1667	0.1666	0.1667
x_3	0.0527	0.0574	0.0540	0.0568	0.0546	0.0563	0.0550	0.0560	0.0552

x_1	0.3891	0.3887	0.3890	0.3888	0.3890	0.3888	0.3889	0.3888	0.3889
x_2	0.1666	0.1667	0.1667	0.1667	0.1667	0.1667	0.1667	0.1667	0.1667
x_3	0.0559	0.0553	0.0557	0.0554	0.0557	0.0555	0.0556	0.0555	0.0556

x_1	0.3889	0.3889							
x_2	0.1667	0.1667							
x_3	0.0555	0.0556							

Gauss-Seidel: Radio espectral – 0.5, número de iteraciones – 10

x_1	0	0.2500	0.3333	0.3958	0.4063	0.3958	0.3880	0.3867	0.3880	0.3890
x_2	0	0.1875	0.2187	0.1875	0.1641	0.1602	0.1641	0.1670	0.1675	0.1670
x_3	0	0.2083	0.1458	0.0625	0.0365	0.0443	0.0547	0.0579	0.0570	0.0557

4.7.28 Jacobi: Radio espectral – 0.5241, número de iteraciones – 18

x_1	0	1.0000	1.1833	1.1302	1.0158	1.0612	1.0836	1.0773	1.0692
x_2	0	0.8000	0.5375	0.1055	0.1792	0.2691	0.2548	0.2249	0.2285
x_3	0	0.3750	−0.1469	−0.0809	−0.0351	−0.0211	−0.0491	−0.0487	−0.0435
x_4	0	−0.1250	0.1469	0.0717	0.0194	0.0200	0.0386	0.0362	0.0317

x_1	1.0715	1.0735	1.0731	1.0725	1.0726	1.0728	1.0728	1.0727	1.0727
x_2	0.2361	0.2355	0.2333	0.2334	0.2340	0.2340	0.2338	0.2338	0.2338
x_3	−0.0426	−0.0444	−0.0446	−0.0442	−0.0441	−0.0442	−0.0442	−0.0442	−0.0442
x_4	0.0318	0.0331	0.0331	0.0327	0.0327	0.0328	0.0328	0.0328	0.0328

Gauss-Seidel: Radio espectral – 0.2571, número de iteraciones – 9

x_1	0	1.0000	1.0914	1.0734	1.0713	1.0730	1.0728	1.0727	1.0727
x_2	0	0.3000	0.2285	0.2296	0.2349	0.2339	0.2338	0.2339	0.2338
x_3	0	−0.0375	−0.0505	−0.0436	−0.0440	−0.0443	−0.0442	−0.0442	−0.0442
x_4	0	0.0328	0.0337	0.0324	0.0328	0.0328	0.0328	0.0328	0.0328

4.7.29 $x_1 = -17.285714,\ x_2 = 10.142857,\ x_3 = -3.928571$ y $x_4 = 1$.

4.7.30 $x_1 = -3.411764,\ x_2 = -0.611764,\ x_3 = 0.647058,\ x_4 = 2.658823$ y $x_5 = 3$.

4.7.31 $x_1 = -11.4,\ x_2 = 0.933333,\ x_3 = 11.733333,\ x_4 = 2$ y $x_5 = 3$.

4.7.32 $x_1 = -9.224880,\ x_2 = -12.071770,\ x_3 = -8.681020,\ x_4 = 19.567783,\ x_5 = 2.778309,\ x_6 = 1$ y $x_7 = 5$.

4.7.33 $x_1 = -0.058075,\ x_2 = 0.405285,\ x_3 = 0.651030$ y $x_4 = -0.163925$.

www.FreeLibros.me

4.7.34 $x_1 = 0.443628$ y $x_2 = 1.003015$.

4.7.35 $x_1 = 0.157609$ y $x_2 = 0.180039$.

4.7.36 $x_1 = -0.643275$, $x_2 = 0.272889$ y $x_3 = 1.172951$.

Capítulo 5

5.10.1 $P_4(x) = -0.039212x^4 - 2.069885x^3 + 19.518582x^2 - 38.911326x + 25.407402$.

x	1.2	1.9	2.1
$f(x)$	3.162497	7.229604	9.838755

5.10.2 $P_3(x) = 0.013622x^3 - 0.131880x^2 + 0.211644x + 0.649802$.

x	2	4
$f(x)$	0.654551	0.258151

5.10.3 $P_3(x) = 0.269647x^3 + 1.053044x^2 - 6.924038x - 7.65$.

5.10.4 $P_2(x) = 1.005555x^2 - 9.272222x + 26.922222$.

5.10.5 $P_3(x) = -0.041896x^3 + 4.671977x^2 - 133.835438x + 2239.657665$.

5.10.6 $P_5(x) = 0.0416x^5 - 42.0833x^4 + 17\,001.125x^3 - 3\,434\,008.4166x^2 + 346\,801\,698.3333x$
$-14\,009\,002\,998$.

5.10.7 $P_7(x) = -845\,130.1680x^7 + 937\,537.3197x^6 - 396\,672.2423x^5 + 80\,815.4463x^4$
$-8\,283.2733x^3 + 434.8406x^2 - 14.4112x + 1.1081$.

5.10.8 $P_4(x) = -0.572916x^4 + 9.645833x^3 - 51.833333x^2 + 115.354166x - 66.59375$.

5.10.9 $P_5(x) = 0.0239x^5 - 0.9817x^4 + 15.1979x^3 - 108.6197x^2 + 349.2781x - 379.8984$.

5.10.10 $P_4(x) = 0.208333x^4 - 2.416666x^3 + 12.291666x^2 - 27.083333x + 26$.

5.10.11 $P_8(x) = -\left(3.044114\times(10)^{-6}\right)x^8 + 0.000217x^7 - 0.006329x^6 + 0.096439x^5$
$-0.828034x^4 + 4.024025x^3 - 10.418702x^2 + 10.890720x + 6.241666$.

5.10.12 $P_6(x) = +\left(2.532021\times(10)^{-6}\right)x^6 - 0.000202x^5 + 0.006068x^4 + 0.084587x^3$
$+0.548857x^2 - 1.084744x + 2.614606$.

5.10.13 $P_3(x) = -3.166666x^3 + 61.5x^2 - 386.333333x + 791$.

5.10.14 $P_2(x) = -0.0234375x^2 + 0.6875x + 1$.

5.10.15 $x_1 = 0$, $x_2 = 6.698729$, $x_3 = 25$, $x_4 = 50$, $x_5 = 75$, $x_6 = 93.301270$ y $x_7 = 100$
$P_6(x) = -\left(9.386666\times(10)^{-9}\right)x^6 + \left(1.542454\times(10)^{-6}\right)x^5 - \left(4.150707\times(10)^{-5}\right)x^4$
$+0.002722x^3 + 0.061157x^2 + 2.110515x + 13$.

5.10.16 $x_1 = 36$, $x_2 = 36.75$, $x_3 = 38.25$ y $x_4 = 39$
$P_3(x) = 0.225185x^3 - 25.582222x^2 + 968.42x - 12214.55$.

5.10.17 $x_1 = 33$, $x_2 = 33.742733$, $x_3 = 35.823826$, $x_4 = 38.831092$, $x_5 = 42.169070$, $x_6 = 45.176173$,
$x_7 = 47.257266$ y $x_8 = 48$.
$P_7(x) = -\left(1.664406\times(10)^{-5}\right)x^7 + 0.004443x^6 - 0.487007x^5 + 28.098524x^4$
$-900.971464x^3 + 15228.790782x^2 - 105765.224663x - 7853.503150$.

5.10.18 $x_1 = 4.257$, $x_2 = 4.617887$, $x_3 = 5.645606$, $x_4 = 7.183697$, $x_5 = 8.998$, $x_6 = 10.812302$,
$x_7 = 12.350393$, $x_8 = 13.378112$ y $x_9 = 13.739$.

5.10.19 $P_3(x) = -0.010703x^3 - 0.111941x^2 + 7.376083x - 9.601248$.

www.FreeLibros.me

5.10.20 $P_4(x) = -0.000333x^4 + 0.015909x^3 - 0.182778x^2 + 2.130693x - 1.211743.$

5.10.21 $P_3(x) = -0.000129x^3 + 0.015748x^2 - 1.233630x + 7.842451.$

5.10.22 $P_2(x) = -0.000880x^2 - 0.217534x + 7.485780,$

$P_3(x) = 8.887015x^3 - 0.004735x^2 - 0.172627x + 7.374392.$

5.10.23 $P_2(x) = 0.014978x^2 - 0.577271x + 30.203384.$

5.10.24 $P_2(x) = 0.061904x^2 - 0.766666x + 3.285714.$

5.10.25 $P_6(x) = 0.000898x^6 - 0.043774x^5 + 0.807770x^4 - 7.033590x^3$

$+ 28.747029x^2 - 40.923950x + 43.782692.$

5.10.26

Función discreta		Descomposición de Fourier con $n = 0, 1, \ldots, 15$	
t	$f(t)$	Funciones coseno	Funciones seno
0	20	$100.8865\cos(0)$	$100.8865\ \mathrm{sen}(0)$
0.2	18.0053	$35.9863\cos(2\pi n\Delta\varphi_1)$	$-46.7374\ \mathrm{sen}(2\pi n\Delta\varphi_1)$
0.4	15.1571	$14.9021\cos(4\pi n\Delta\varphi_2)$	$-25.8392\ \mathrm{sen}(4\pi n\Delta\varphi_2)$
0.6	12.2633	$11.6751\cos(6\pi n\Delta\varphi_3)$	$-15.7561\ \mathrm{sen}(6\pi n\Delta\varphi_3)$
0.8	9.6219	$10.8065\cos(8\pi n\Delta\varphi_4)$	$-10.3925\ \mathrm{sen}(8\pi n\Delta\varphi_4)$
1.0	7.3575	$10.4761\cos(10\pi n\Delta\varphi_5)$	$-6.8916\ \mathrm{sen}(10\pi n\Delta\varphi_5)$
1.2	5.5010	$10.3281\cos(12\pi n\Delta\varphi_6)$	$-4.2537\ \mathrm{sen}(12\pi n\Delta\varphi_6)$
1.4	4.0311	$10.2613\cos(14\pi n\Delta\varphi_7)$	$-2.0381\ \mathrm{sen}(14\pi n\Delta\varphi_7)$
1.6	2.9002	$10.2419\cos(16\pi n\Delta\varphi_8)$	$0.0000\ \mathrm{sen}(16\pi n\Delta\varphi_8)$
1.8	2.0516	$10.2613\cos(18\pi n\Delta\varphi_9)$	$2.0381\ \mathrm{sen}(18\pi n\Delta\varphi_9)$
2.0	1.4286	$10.3281\cos(20\pi n\Delta\varphi_{10})$	$4.2537\ \mathrm{sen}(20\pi n\Delta\varphi_{10})$
2.2	0.9802	$10.4761\cos(22\pi n\Delta\varphi_{11})$	$6.8916\ \mathrm{sen}(22\pi n\Delta\varphi_{11})$
2.4	0.6632	$10.8065\cos(24\pi n\Delta\varphi_{12})$	$10.3925\ \mathrm{sen}(24\pi n\Delta\varphi_{12})$
2.6	0.4428	$11.6751\cos(26\pi n\Delta\varphi_{13})$	$15.7561\ \mathrm{sen}(26\pi n\Delta\varphi_{13})$
2.8	0.2919	$14.9021\cos(28\pi n\Delta\varphi_{14})$	$25.8392\ \mathrm{sen}(28\pi n\Delta\varphi_{14})$
3.0	0.1901	$35.9863\cos(30\pi n\Delta\varphi_{15})$	$46.7374\ \mathrm{sen}(30\pi n\Delta\varphi_{15})$

www.FreeLibros.me

5.10.27

Función discreta		Descomposición de Fourier con $n = 0, 1, \ldots, 31$	
t	$f(t)$	Funciones coseno	Funciones seno
0	0	$7.8888\cos(0)$	$7.8888\,\text{sen}(0)$
1/155	0.003452	$-2.4581\cos(2\pi n\Delta\varphi_1)$	$-6.2283\,\text{sen}(2\pi n\Delta\varphi_1)$
2/155	0.016987	$5.7960\cos(4\pi n\Delta\varphi_2)$	$7.1313\,\text{sen}(4\pi n\Delta\varphi_2)$
3/155	0.043114	$-3.5792\cos(6\pi n\Delta\varphi_3)$	$3.6293\,\text{sen}(6\pi n\Delta\varphi_3)$
4/155	0.083407	$-2.3890\cos(8\pi n\Delta\varphi_4)$	$-0.1345\,\text{sen}(8\pi n\Delta\varphi_4)$
1/31	0.138927	$-1.0506\cos(10\pi n\Delta\varphi_5)$	$-0.6270\,\text{sen}(10\pi n\Delta\varphi_5)$
6/155	0.210201	$-0.4803\cos(12\pi n\Delta\varphi_6)$	$-0.5400\,\text{sen}(12\pi n\Delta\varphi_6)$
7/155	0.297036	$-0.2260\cos(14\pi n\Delta\varphi_7)$	$-0.4240\,\text{sen}(14\pi n\Delta\varphi_7)$
8/155	0.398181	$-0.0946\cos(16\pi n\Delta\varphi_8)$	$-0.3340\,\text{sen}(16\pi n\Delta\varphi_8)$
9/155	0.510886	$-0.0179\cos(18\pi n\Delta\varphi_9)$	$-0.2653\,\text{sen}(18\pi n\Delta\varphi_9)$
2/31	0.630368	$0.0303\cos(20\pi n\Delta\varphi_{10})$	$-0.2109\,\text{sen}(20\pi n\Delta\varphi_{10})$
11/155	0.749281	$0.0624\cos(22\pi n\Delta\varphi_{11})$	$-0.1657\,\text{sen}(22\pi n\Delta\varphi_{11})$
12/155	0.857294	$0.0841\cos(24\pi n\Delta\varphi_{12})$	$-0.1268\,\text{sen}(24\pi n\Delta\varphi_{12})$
13/155	0.940994	$0.0988\cos(26\pi n\Delta\varphi_{13})$	$-0.0920\,\text{sen}(26\pi n\Delta\varphi_{13})$
14/155	0.984353	$0.1083\cos(28\pi n\Delta\varphi_{14})$	$-0.0600\,\text{sen}(28\pi n\Delta\varphi_{14})$
3/31	0.970083	$0.1136\cos(30\pi n\Delta\varphi_{15})$	$-0.0296\,\text{sen}(30\pi n\Delta\varphi_{15})$
16/155	0.882140	$0.1153\cos(32\pi n\Delta\varphi_{16})$	$0.0000\,\text{sen}(32\pi n\Delta\varphi_{16})$
17/155	0.709521	$0.1136\cos(34\pi n\Delta\varphi_{17})$	$0.0296\,\text{sen}(34\pi n\Delta\varphi_{17})$
18/155	0.451163	$0.1083\cos(36\pi n\Delta\varphi_{18})$	$0.0600\,\text{sen}(36\pi n\Delta\varphi_{18})$
19/155	0.121189	$0.0988\cos(38\pi n\Delta\varphi_{19})$	$0.0920\,\text{sen}(38\pi n\Delta\varphi_{19})$
4/31	−0.246924	$0.0841\cos(40\pi n\Delta\varphi_{20})$	$0.1268\,\text{sen}(40\pi n\Delta\varphi_{20})$
21/155	−0.599445	$0.0624\cos(42\pi n\Delta\varphi_{21})$	$0.1657\,\text{sen}(42\pi n\Delta\varphi_{21})$
22/155	−0.867265	$0.0303\cos(44\pi n\Delta\varphi_{22})$	$0.2109\,\text{sen}(44\pi n\Delta\varphi_{22})$
23/155	−0.978610	$-0.0179\cos(46\pi n\Delta\varphi_{23})$	$0.2653\,\text{sen}(46\pi n\Delta\varphi_{23})$
24/155	−0.879664	$-0.0946\cos(48\pi n\Delta\varphi_{24})$	$0.3340\,\text{sen}(48\pi n\Delta\varphi_{24})$
5/31	−0.559677	$-0.2260\cos(50\pi n\Delta\varphi_{25})$	$0.4240\,\text{sen}(50\pi n\Delta\varphi_{25})$
26/155	−0.072425	$-0.4803\cos(52\pi n\Delta\varphi_{26})$	$0.5400\,\text{sen}(52\pi n\Delta\varphi_{26})$
27/155	0.458013	$-1.0506\cos(54\pi n\Delta\varphi_{27})$	$0.6270\,\text{sen}(54\pi n\Delta\varphi_{27})$
28/155	0.859303	$-2.3890\cos(56\pi n\Delta\varphi_{28})$	$0.1345\,\text{sen}(56\pi n\Delta\varphi_{28})$
29/155	0.965699	$-3.5792\cos(58\pi n\Delta\varphi_{29})$	$-3.6293\,\text{sen}(58\pi n\Delta\varphi_{29})$
6/31	0.694967	$5.7960\cos(60\pi n\Delta\varphi_{30})$	$-7.1313\,\text{sen}(60\pi n\Delta\varphi_{30})$
1/5	0.116250	$-2.4581\cos(62\pi n\Delta\varphi_{31})$	$6.2283\,\text{sen}(62\pi n\Delta\varphi_{31})$

www.FreeLibros.me

5.10.28

Función discreta			Descomposición de Fourier con $n=0,1,\ldots,31$	
t	$f(t)$		Funciones coseno	Funciones seno
0	0		$5.8927\cos(0)$	$5.8927\,\mathrm{sen}(0)$
7/31	0.024984		$-0.9800\cos(2\,\pi\,n\Delta\varphi_1)$	$-3.7144\,\mathrm{sen}(2\,\pi\,n\Delta\varphi_1)$
14/31	0.104023		$-1.4643\cos(4\,\pi\,n\Delta\varphi_2)$	$-0.0354\,\mathrm{sen}(4\,\pi\,n\Delta\varphi_2)$
21/31	0.213349		$-0.3985\cos(6\,\pi\,n\Delta\varphi_3)$	$0.3466\,\mathrm{sen}(6\,\pi\,n\Delta\varphi_3)$
28/31	0.324830		$-0.0935\cos(8\,\pi\,n\Delta\varphi_4)$	$0.1974\,\mathrm{sen}(8\,\pi\,n\Delta\varphi_4)$
35/31	0.418132		$-0.0208\cos(10\,\pi\,n\Delta\varphi_5)$	$0.1026\,\mathrm{sen}(10\,\pi\,n\Delta\varphi_5)$
42/31	0.482525		$-0.0027\cos(12\,\pi\,n\Delta\varphi_6)$	$0.0564\,\mathrm{sen}(12\,\pi\,n\Delta\varphi_6)$
49/31	0.515260		$0.0016\cos(14\,\pi\,n\Delta\varphi_7)$	$0.0564\,\mathrm{sen}(14\,\pi\,n\Delta\varphi_7)$
56/31	0.518969		$0.0024\cos(16\,\pi\,n\Delta\varphi_8)$	$0.0207\,\mathrm{sen}(16\,\pi\,n\Delta\varphi_8)$
63/31	0.499220		$0.0022\cos(18\,\pi\,n\Delta\varphi_9)$	$0.0136\,\mathrm{sen}(18\,\pi\,n\Delta\varphi_9)$
70/31	0.462630		$0.0018\cos(20\,\pi\,n\Delta\varphi_{10})$	$0.0092\,\mathrm{sen}(20\,\pi\,n\Delta\varphi_{10})$
77/31	0.415603		$0.0014\cos(22\,\pi\,n\Delta\varphi_{11})$	$0.0063\,\mathrm{sen}(22\,\pi\,n\Delta\varphi_{11})$
84/31	0.363627		$0.0011\cos(24\,\pi\,n\Delta\varphi_{12})$	$0.0043\,\mathrm{sen}(24\,\pi\,n\Delta\varphi_{12})$
91/31	0.310976		$0.0009\cos(26\,\pi\,n\Delta\varphi_{13})$	$0.0029\,\mathrm{sen}(26\,\pi\,n\Delta\varphi_{13})$
98/31	0.260688		$0.0007\cos(28\,\pi\,n\Delta\varphi_{14})$	$0.0018\,\mathrm{sen}(28\,\pi\,n\Delta\varphi_{14})$
105/31	0.214698		$0.0006\cos(30\,\pi\,n\Delta\varphi_{15})$	$0.0008\,\mathrm{sen}(30\,\pi\,n\Delta\varphi_{15})$
112/31	0.174042		$0.0006\cos(32\,\pi\,n\Delta\varphi_{16})$	$0.0000\,\mathrm{sen}(32\,\pi\,n\Delta\varphi_{16})$
119/31	0.139081		$0.0006\cos(34\,\pi\,n\Delta\varphi_{17})$	$-0.0008\,\mathrm{sen}(34\,\pi\,n\Delta\varphi_{17})$
126/31	0.109706		$0.0007\cos(36\,\pi\,n\Delta\varphi_{18})$	$-0.0018\,\mathrm{sen}(36\,\pi\,n\Delta\varphi_{18})$
133/31	0.085511		$0.0009\cos(38\,\pi\,n\Delta\varphi_{19})$	$-0.0029\,\mathrm{sen}(38\,\pi\,n\Delta\varphi_{19})$
140/31	0.065925		$0.0011\cos(40\,\pi\,n\Delta\varphi_{20})$	$-0.0043\,\mathrm{sen}(40\,\pi\,n\Delta\varphi_{20})$
147/31	0.050311		$0.0014\cos(42\,\pi\,n\Delta\varphi_{21})$	$-0.0063\,\mathrm{sen}(42\,\pi\,n\Delta\varphi_{21})$
154/31	0.038034		$0.0018\cos(44\,\pi\,n\Delta\varphi_{22})$	$-0.0092\,\mathrm{sen}(44\,\pi\,n\Delta\varphi_{22})$
161/31	0.028501		$0.0022\cos(46\,\pi\,n\Delta\varphi_{23})$	$-0.0136\,\mathrm{sen}(46\,\pi\,n\Delta\varphi_{23})$
168/31	0.021181		$0.0024\cos(48\,\pi\,n\Delta\varphi_{24})$	$-0.0207\,\mathrm{sen}(48\,\pi\,n\Delta\varphi_{24})$
175/31	0.015619		$0.0016\cos(50\,\pi\,n\Delta\varphi_{25})$	$-0.0332\,\mathrm{sen}(50\,\pi\,n\Delta\varphi_{25})$
182/31	0.011433		$-0.0027\cos(52\,\pi\,n\Delta\varphi_{26})$	$-0.0564\,\mathrm{sen}(52\,\pi\,n\Delta\varphi_{26})$
189/31	0.008311		$-0.0208\cos(54\,\pi\,n\Delta\varphi_{27})$	$-0.1026\,\mathrm{sen}(54\,\pi\,n\Delta\varphi_{27})$
196/31	0.006002		$-0.0935\cos(56\,\pi\,n\Delta\varphi_{28})$	$-0.1974\,\mathrm{sen}(56\,\pi\,n\Delta\varphi_{28})$
203/31	0.004307		$-0.3985\cos(58\,\pi\,n\Delta\varphi_{29})$	$-0.3466\,\mathrm{sen}(58\,\pi\,n\Delta\varphi_{29})$
210/31	0.003072		$-1.4643\cos(60\,\pi\,n\Delta\varphi_{30})$	$0.0354\,\mathrm{sen}(60\,\pi\,n\Delta\varphi_{30})$
7	0.002179		$-0.9800\cos(62\,\pi\,n\Delta\varphi_{31})$	$3.7144\,\mathrm{sen}(62\,\pi\,n\Delta\varphi_{31})$

www.FreeLibros.me

5.10.29

Función discreta		Descomposición de Fourier con $n = 0, 1, \ldots, 7$	
t	$f(t)$	Funciones coseno	Funciones seno
0	1.12	$11.59\cos(0)$	$11.59\,\text{sen}(0)$
0.2	1.16	$-1.5358\cos(2\,\pi\, n\Delta\varphi_1)$	$-0.2302\,\text{sen}(2\,\pi\, n\Delta\varphi_1)$
0.4	1.45	$0.3\cos(4\,\pi\, n\Delta\varphi_2)$	$0.27\,\text{sen}(4\,\pi\, n\Delta\varphi_2)$
0.6	1.78	$-0.1641\cos(6\,\pi\, n\Delta\varphi_3)$	$-0.0102\,\text{sen}(6\,\pi\, n\Delta\varphi_3)$
0.8	1.97	$0.17\cos(8\,\pi\, n\Delta\varphi_4)$	$0.0000\,\text{sen}(8\,\pi\, n\Delta\varphi_4)$
1.0	1.56	$-0.1641\cos(10\,\pi\, n\Delta\varphi_5)$	$0.0102\,\text{sen}(10\,\pi\, n\Delta\varphi_5)$
1.2	1.34	$0.3\cos(12\,\pi\, n\Delta\varphi_6)$	$-0.27\,\text{sen}(12\,\pi\, n\Delta\varphi_6)$
1.4	1.21	$-1.5358\cos(14\,\pi\, n\Delta\varphi_7)$	$0.2302\,\text{sen}(14\,\pi\, n\Delta\varphi_7)$

5.10.30

Función discreta		Descomposición de Fourier con $n = 0, 1, \ldots, 15$	
t	$f(t)$	Funciones coseno	Funciones seno
0	3.45	$80.11\cos(0)$	$80.11\,\text{sen}(0)$
0.1	5.35	$2.1402\cos(2\,\pi\, n\Delta\varphi_1)$	$-3.0657\,\text{sen}(2\,\pi\, n\Delta\varphi_1)$
0.2	6.61	$-3.9251\cos(4\,\pi\, n\Delta\varphi_2)$	$1.1876\,\text{sen}(4\,\pi\, n\Delta\varphi_2)$
0.3	3.45	$4.1224\cos(6\,\pi\, n\Delta\varphi_3)$	$-0.8108\,\text{sen}(6\,\pi\, n\Delta\varphi_3)$
0.4	6.95	$-2.96\cos(8\,\pi\, n\Delta\varphi_4)$	$1.59\,\text{sen}(8\,\pi\, n\Delta\varphi_4)$
0.5	5.25	$-3.9122\cos(10\,\pi\, n\Delta\varphi_5)$	$0.7347\,\text{sen}(10\,\pi\, n\Delta\varphi_5)$
0.6	6.32	$-7.4748\cos(12\,\pi\, n\Delta\varphi_6)$	$4.9076\,\text{sen}(12\,\pi\, n\Delta\varphi_6)$
0.7	4.28	$-5.5504\cos(14\,\pi\, n\Delta\varphi_7)$	$0.4797\,\text{sen}(14\,\pi\, n\Delta\varphi_7)$
0.8	4.25	$10.21\cos(16\,\pi\, n\Delta\varphi_8)$	$0.0000\,\text{sen}(16\,\pi\, n\Delta\varphi_8)$
0.9	2.54	$-5.5504\cos(18\,\pi\, n\Delta\varphi_9)$	$-0.4797\,\text{sen}(18\,\pi\, n\Delta\varphi_9)$
1.0	6.35	$-7.4748\cos(20\,\pi\, n\Delta\varphi_{10})$	$-4.9076\,\text{sen}(20\,\pi\, n\Delta\varphi_{10})$
1.1	3.98	$-3.9122\cos(22\,\pi\, n\Delta\varphi_{11})$	$-0.7347\,\text{sen}(22\,\pi\, n\Delta\varphi_{11})$
1.2	6.45	$-2.96\cos(24\,\pi\, n\Delta\varphi_{12})$	$-1.59\,\text{sen}(24\,\pi\, n\Delta\varphi_{12})$
1.3	3.54	$4.1224\cos(26\,\pi\, n\Delta\varphi_{13})$	$0.8108\,\text{sen}(26\,\pi\, n\Delta\varphi_{13})$
1.4	4.78	$-3.9251\cos(28\,\pi\, n\Delta\varphi_{14})$	$-1.1876\,\text{sen}(28\,\pi\, n\Delta\varphi_{14})$
1.5	6.56	$2.1402\cos(30\,\pi\, n\Delta\varphi_{15})$	$3.0657\,\text{sen}(30\,\pi\, n\Delta\varphi_{15})$

5.10.31 $T_2(x) = 2x^2 - 1,\ T_3(x) = 4x^3 - 3x$ y $T_4(x) = 8x^4 - 8x^2 + 1.$

5.10.32 $Tchebyshev_4(x) = 0.177347x^3 + 0.553939x^2 + 0.997853x + 0.989141.$

5.10.33 $Tchebyshev_6(x) = 0.042774x^5 - 0.071549x^4 + 0.051054x^3$

$$-0.115205x^2 + 0.243195x + 1.056851.$$

www.FreeLibros.me

5.10.34 $Tchebyshev_{10}(x) = 346.3065x^9 - (1.2221e^{-11})x^8 - 572.5715x^7 + (2.4542e^{-11})x^6 + 302.8359x^5 - (1.5415e^{-11})x^4 - 46.0764x^3 + (3.1024e^{-12})x^2 + 1.2335x - (9.7477e^{-14})$.

5.10.35 $Tchebyshev_{10}(x) = -1372.9378x^9 + 644.8697x^8 + 3902.5267x^7 - 2027.4859x^6 - 3939.2913x^5 + 2392.3929x^4 + 1609.9416x^3 - 1279.3371x^2 - 191.6263x + 278.1731$.

Capítulo 6

6.9.3 $I = 3.038615$.

6.9.4 $I = 2.110806$.

6.9.5

h	1	1/2	1/4	1/8	1/16	1/32	1/64	1/128
I	3.4817	2.2663	1.7451	1.5060	1.3918	1.3361	1.3086	1.2949

h	1/256	1/512	1/1 024	1/2 048	1/4 096	1/8 192	1/16 384	1/32 768
I	1.2880	1.2846	1.2829	1.2821	1.2817	1.2815	1.2813	1.2813

6.9.6

h	$\pi/2 = 355/226$	355/452	355/904	355/1 808	355/3 616
I	1.483405e−17	0.033633	0.038640	0.040124	0.040497

h	355/7 232	355/14 464	158/12 875	79/12 875	79/25 750
I	0.040590	0.040613	0.040619	0.040620	0.040620

6.9.7

h	3	3/2	3/4	3/8	3/16	3/32
I	82.344250	39.393939	22.036671	15.029053	11.984509	10.579681

h	3/64	3/128	3/256	3/512	3/1 024	3/2 048	3/4 096
I	9.906794	9.577742	9.415064	9.334188	9.293865	9.273733	9.263674

h	3/8 192	3/16 384	3/32 768	3/65 536	3/131 072	3/262 144	3/524 288
I	9.258646	9.256132	9.254876	9.254248	9.253933	9.253776	9.253698

6.9.8 $I = 17.401709$.

6.9.9 $I = 71.439330$.

6.9.10

h	5	5/2	5/4	5/8	5/16	5/32	5/64	5/128
I	0	0.5130	1.2192	1.7051	1.9028	1.9617	1.9774	1.9815

h	5/256	5/512	5/1 024	5/2 048	5/4 096
I	1.9826	1.9829	1.9830	1.9831	1.9831

www.FreeLibros.me

6.9.11

h	5	5/2	5/4	5/8	5/16	5/32	5/64	5/128
I	0	0.2570	0.4647	0.5481	0.5732	0.5803	0.5823	0.5829

h	5/256	5/512	5/1 024
I	0.5831	0.5832	0.5832

6.9.12

h	1/10	1/20	1/40	1/80	1/160	1/320	1/640
I	0.15	0.149999	0.050000	0.049994	0.050001	0.050002	0.050002

6.9.13 $I = 0.101196.$

6.9.14 $I = 88.$

6.9.15

h	1/10	1/20	1/40	1/80	1/160	1/320	1/640
I	0.1004	0.1003	0.1003	0.1003	0.1003	0.1000	0.0999

6.9.16

h	11/5	11/10	11/20	11/40	11/80	11/160
I	26.936087	17.755307	14.514280	13.356146	12.966810	12.849137

h	11/320	11/640	11/1 280	11/2 560	11/5 120	11/10 240
I	12.816948	12.808640	12.806545	12.806019	12.805888	12.805855

6.9.17

h	3/20	3/40	3/80	3/160	3/320	3/640
I	0.064355	0.035397	0.022369	0.017030	0.015108	0.014511

h	3/1 280	3/2 560	3/5 120	3/10 240	3/20 480
I	0.014346	0.014304	0.014293	0.014290	0.014289

6.9.18 $I = 0.002298.$

6.9.19 $I = 54.833333$

6.9.20

h	1	1/2	1/4	1/8	1/16	1/32	1/64
I	11.570670	11.619883	11.624650	11.625009	11.625033	11.625035	11.625035

6.9.21

h	6	3	3/2	3/4	3/8
I	2844	900	778.5	770.90625	770.431640

h	3/16	3/32	3/64
I	770.401977	770.400123	770.400007

www.FreeLibros.me

6.9.22

h	1/2	1/4	1/8	1/16	1/32
I	1.521525	1.523893	1.524808	0.826108	0.993003

h	1/64	1/128	1/256	1/512
I	0.993405	0.993423	0.993424	0.993424

6.9.23 $I = -0.875800$.

6.9.24 $I = 93.75$.

6.9.25

h	2/3	1/3	1/6	1/12	1/24
I	6.227014	6.960032	6.049684	6.375919	6.191791

h	1/48	1/96	1/192	1/384
I	6.189096	6.188959	6.188951	6.188950

6.9.26

h	1	1/2	1/4	1/8	1/16	1/32	1/64
I	25.999477	25.941541	25.934894	25.934380	25.934346	25.934344	25.934344

6.9.27

h	1/3	1/6	1/12	1/24	1/48
I	3.223667	3.223949	3.223970	3.223971	3.223971

6.9.28

h	3	3/2
I	−5.249999	−5.249999

6.9.29

h	7	7/2	7/8	7/64	7/1 024	7/32 768
I	0.405912	−0.563777	−0.054993	−0.049587	−0.049583	−0.049583

6.9.30

h	4	2	1/2	1/16	1/256
I	32.110785	32.205009	32.213385	32.213431	32.213431

6.9.31

h	1	1/2	1/8	1/64
I	0.089117	0.088986	0.088983	0.088983

6.9.32

h	2	1	1/4	1/32
I	2.276318	2.275977	2.275973	2.275973

www.FreeLibros.me

6.9.33

h	4	2	1/2	1/16
I	2.401965	2.400535	2.400558	2.400558

6.9.34

−1.233700	0	0	0	0
−0.616850	−0.411233	0	0	0
−0.526514	−0.496402	−0.502080	0	0
−0.506475	−0.499795	−0.500021	−0.499989	0
−0.501609	−0.499987	−0.500000	−0.499999	−0.500000

6.9.35

5.395751	0	0	0	0	0	
3.480880	2.842590	0	0	0	0	
2.699098	2.438504	2.411565	0	0	0	
2.419987	2.326950	2.319513	2.318052	0	0	
2.333963	2.305289	2.303844	2.303596	2.303539	0	
2.310353	2.302483	2.302296	2.302271	2.302266	2.302265	
2.304260	2.302229	2.302212	2.302211	2.302211	2.302211	2.302211

6.9.36

−3.394413	0	0	0	0
−1.526644	−0.904054	0	0	0
−1.145641	−1.018640	−1.026279	0	0
−1.054233	−1.023764	−1.024106	−1.024071	0
−1.031595	−1.024049	−1.024068	−1.024067	−1.024067

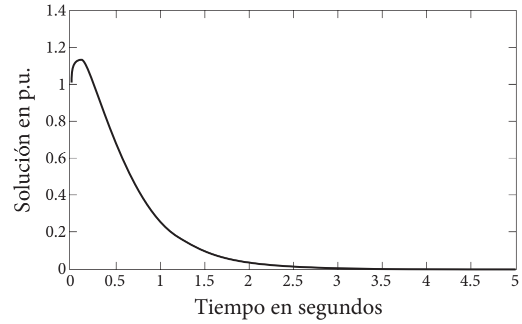
6.9.37

199.491133	0	0	0	0	0
99.745566	66.497044	0	0	0	0
49.900163	33.285029	31.070894	0	0	0
24.042097	15.422742	14.231923	13.964637	0	0
12.666701	8.874903	8.438380	8.346419	8.324387	0
9.550244	8.511425	8.487193	8.487968	8.488523	8.488683

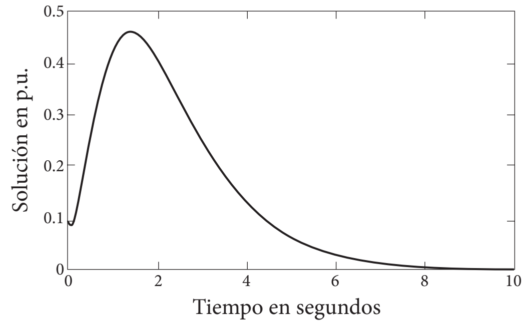
www.FreeLibros.me

Capítulo 7

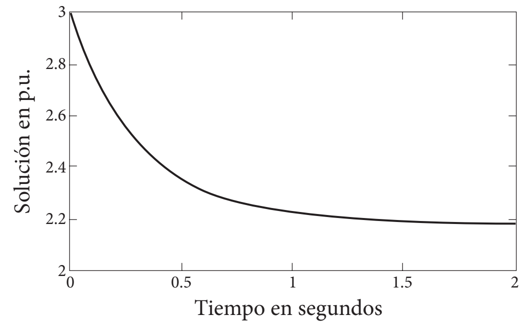
7.10.2 La gráfica de la solución es:



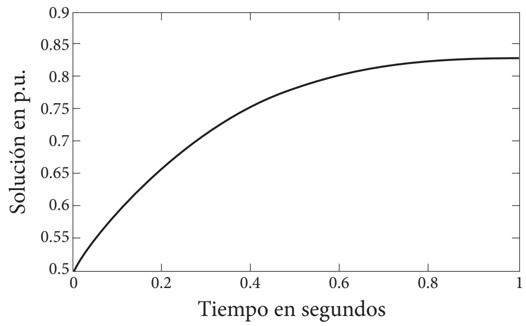
7.10.3 La gráfica de la solución es:



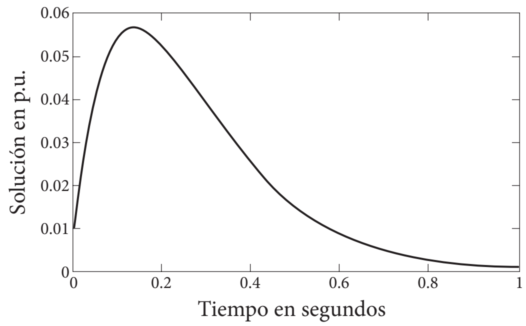
7.10.4 La gráfica de la solución es:



7.10.5 La gráfica de la solución es:

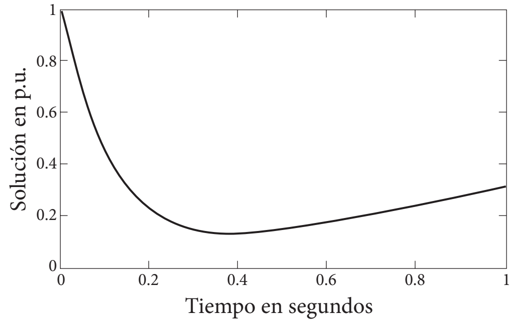


7.10.6 La gráfica de la solución es:

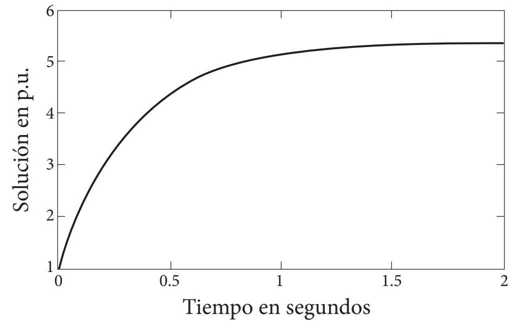


www.FreeLibros.me

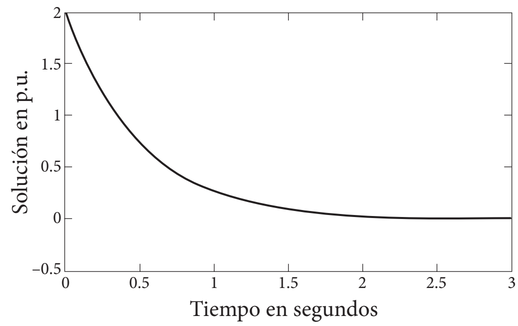
7.10.7 La gráfica de la solución es:



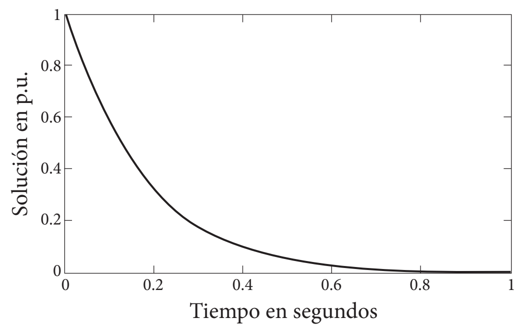
7.10.8 La gráfica de la solución es:



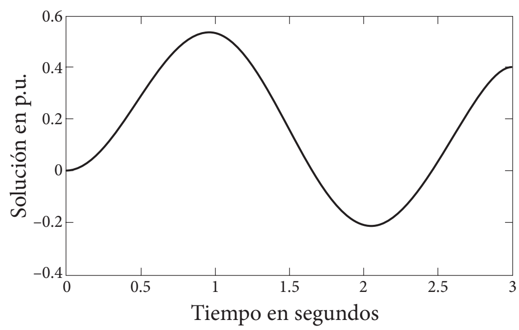
7.10.9 La gráfica de la solución es:



7.10.10 La gráfica de la solución es:

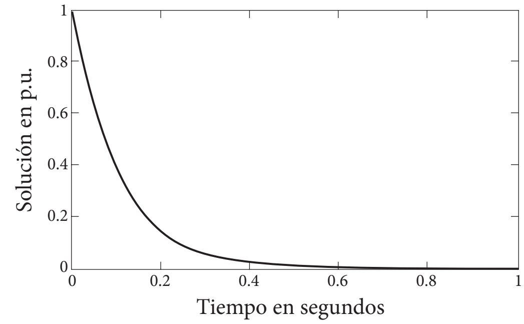


7.10.11 La gráfica de la solución es:

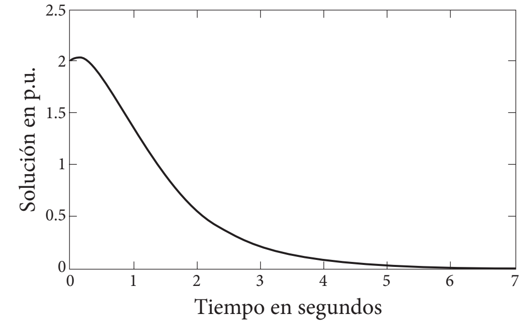


www.FreeLibros.me

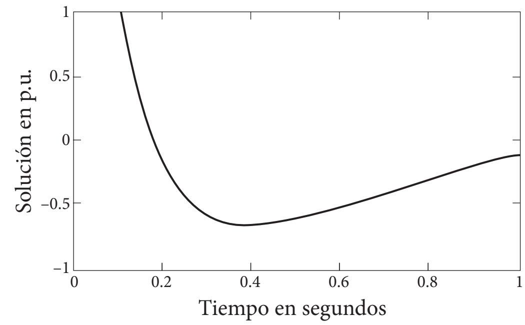
7.10.12 La gráfica de la solución es:



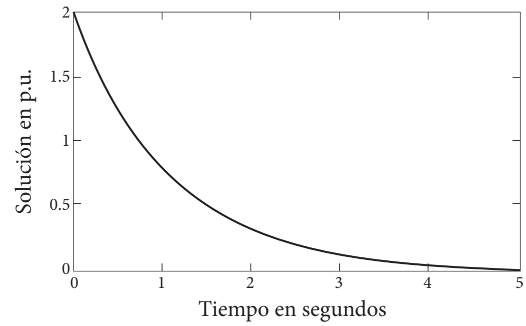
7.10.13 La gráfica de la solución es:



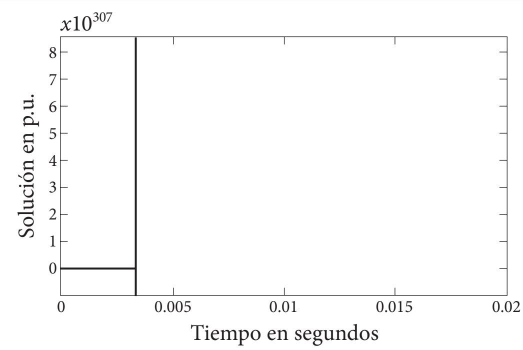
7.10.14 La gráfica de la solución es:



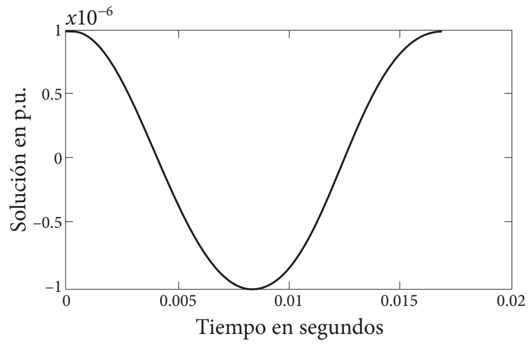
7.10.15 La gráfica de la solución es:



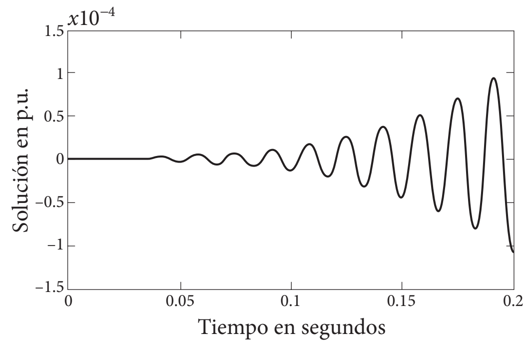
7.10.16 La gráfica de las soluciones es:



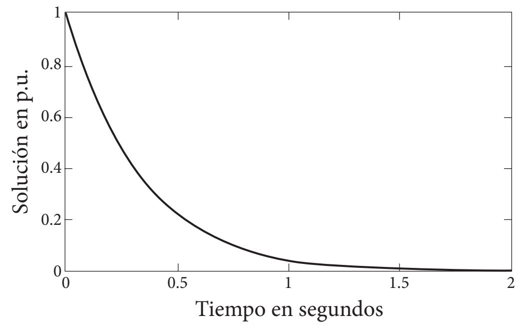
www.FreeLibros.me



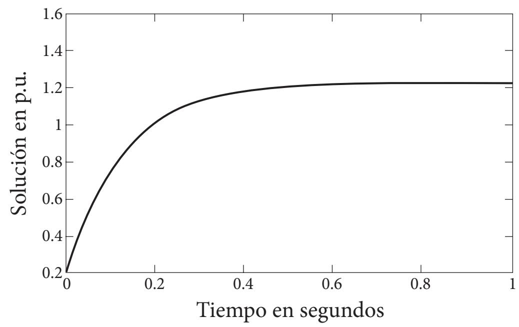
7.10.17 La gráfica de la solución es:



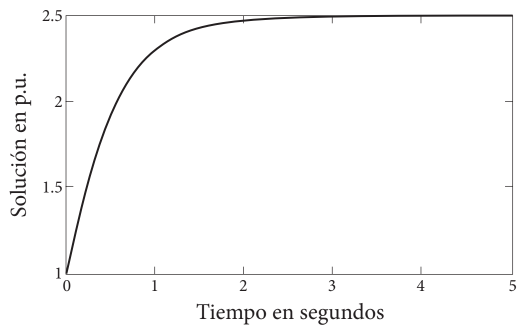
7.10.18 La gráfica de la solución es:



7.10.19 La gráfica de la solución es:

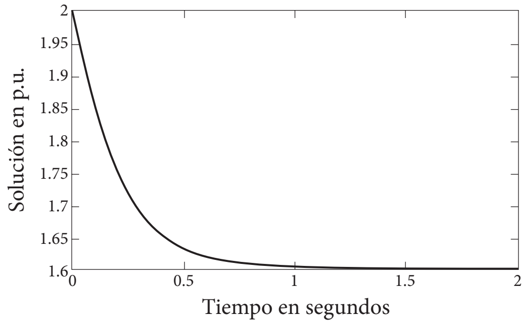


7.10.20 La gráfica de la solución es:

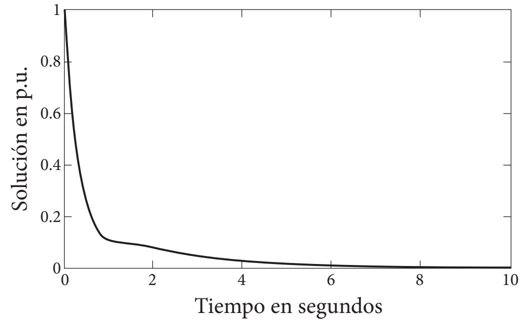


www.FreeLibros.me

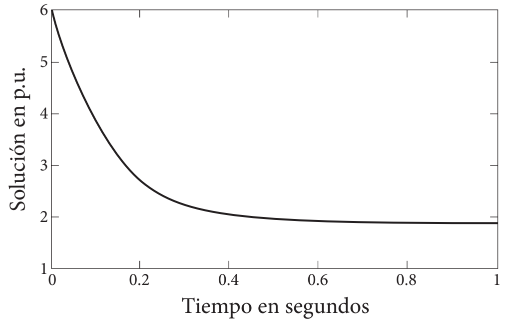
7.10.21 La gráfica de la solución es:



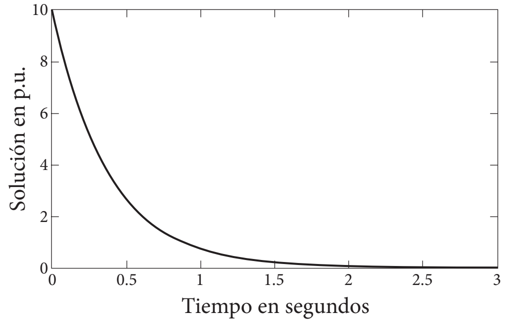
7.10.22 La gráfica de la solución es:



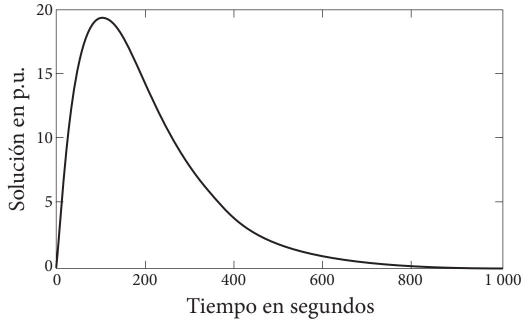
7.10.23 La gráfica de la solución es:



7.10.24 La gráfica de la solución es:

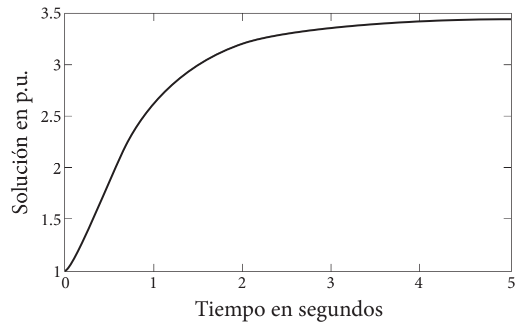


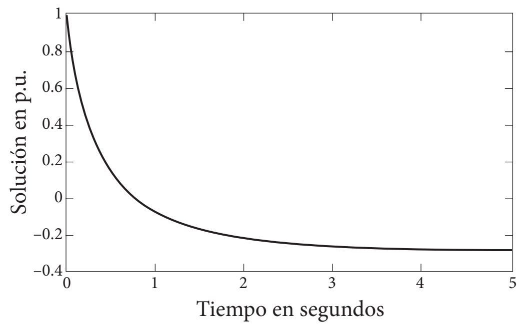
7.10.25 La gráfica de la solución es:



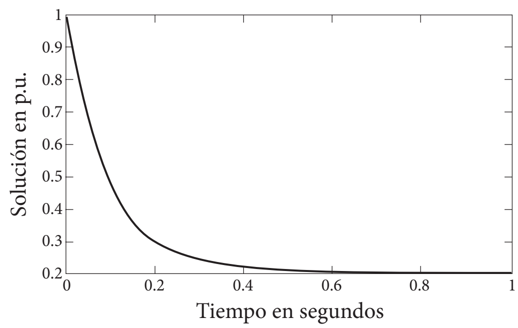
www.FreeLibros.me

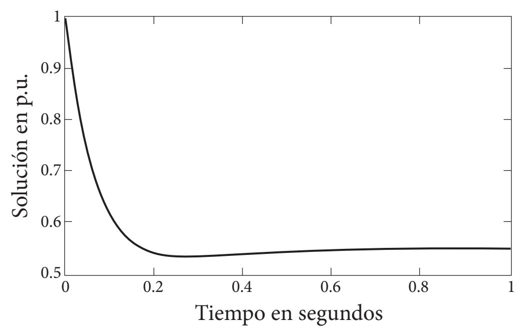
7.10.26 La gráfica de las soluciones es:

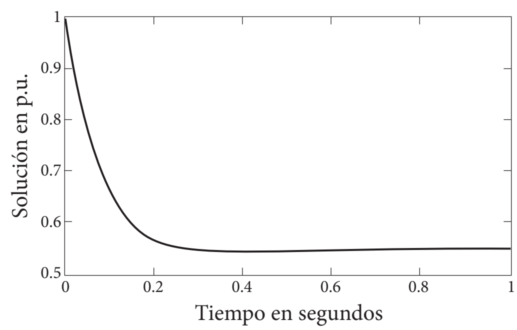




7.10.27 La gráfica de las soluciones es:

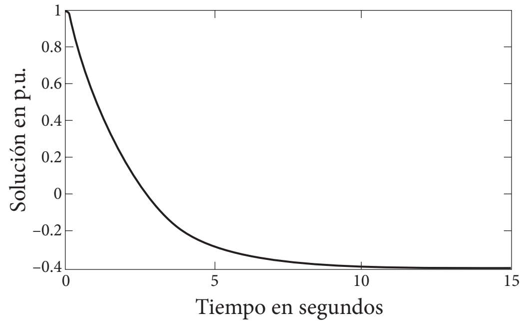


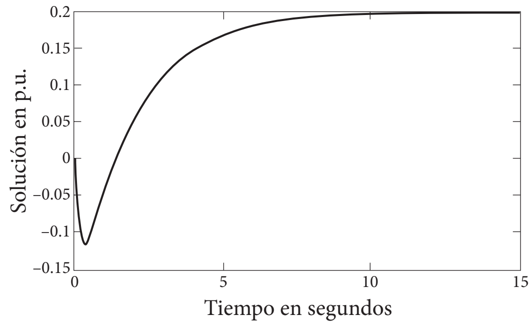


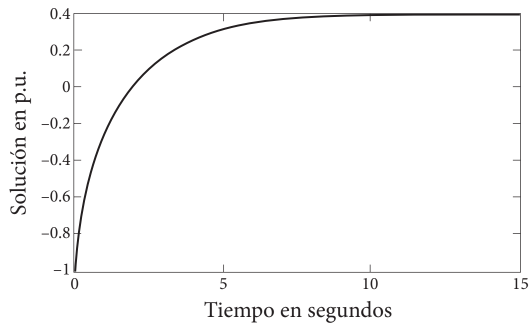


www.FreeLibros.me

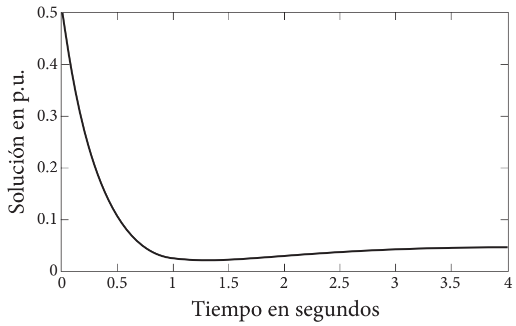
7.10.28 La gráfica de las soluciones es:

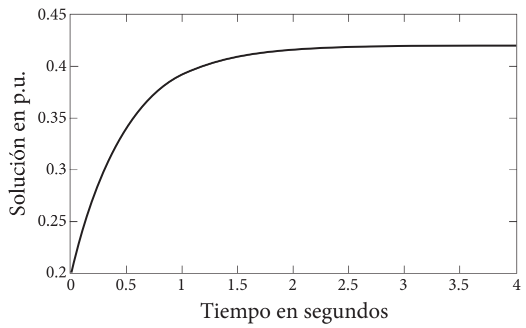




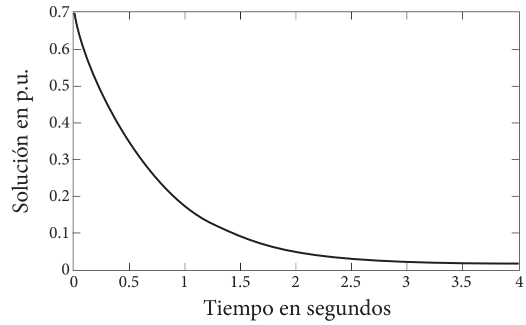


7.10.29 La gráfica de las soluciones es:

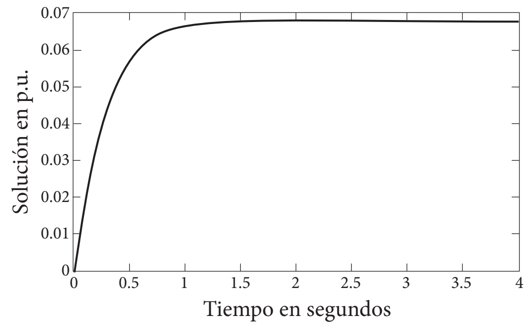


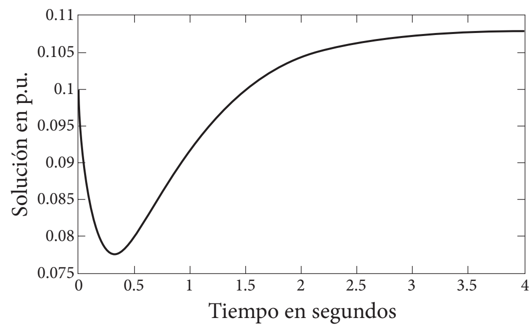


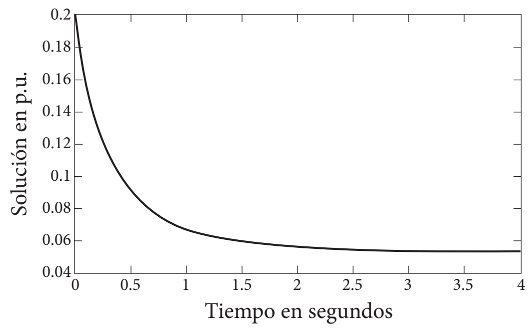
www.FreeLibros.me



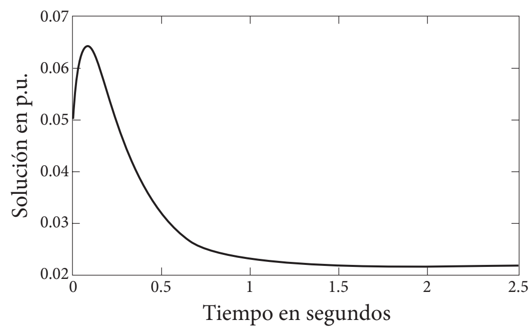
7.10.30 La gráfica de las soluciones es:



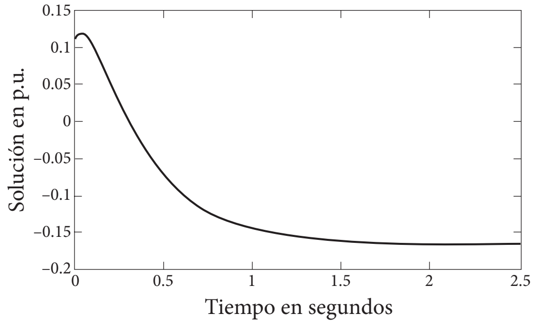


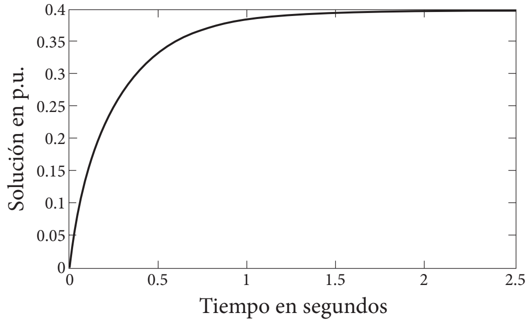


7.10.31 La gráfica de las soluciones es:

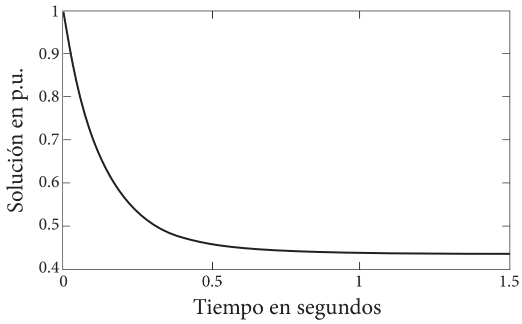


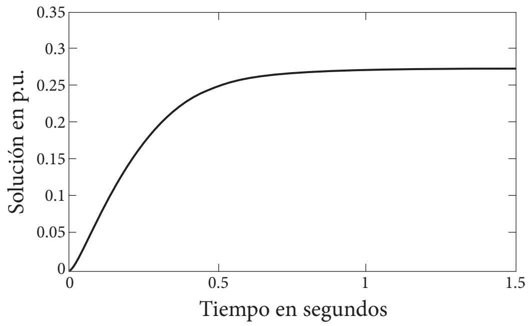
www.FreeLibros.me

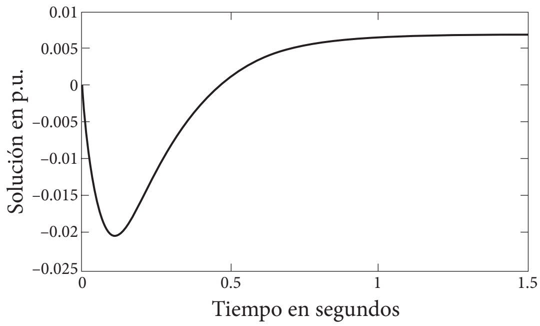




7.10.32 La gráfica de las soluciones es:

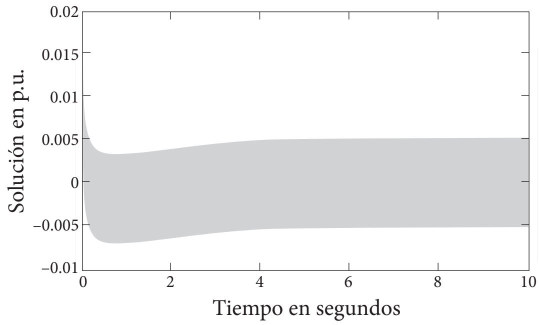


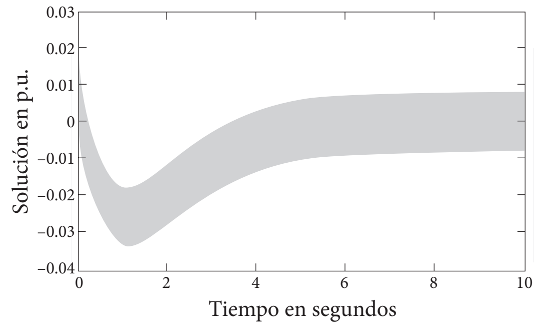


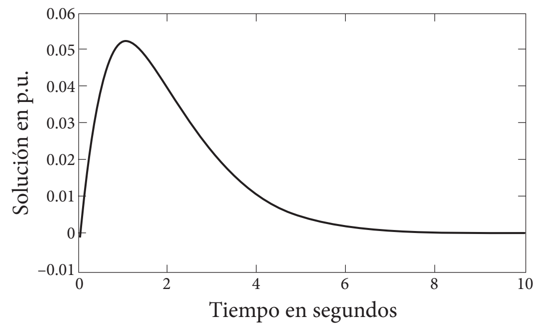


www.FreeLibros.me

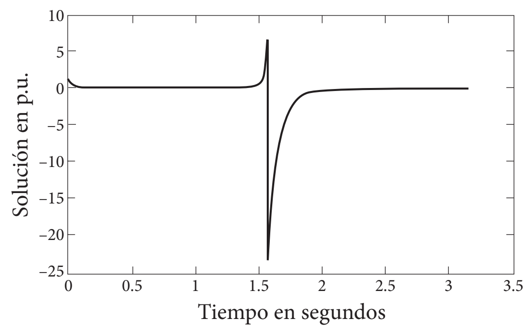
7.10.33 La gráfica de las soluciones es:

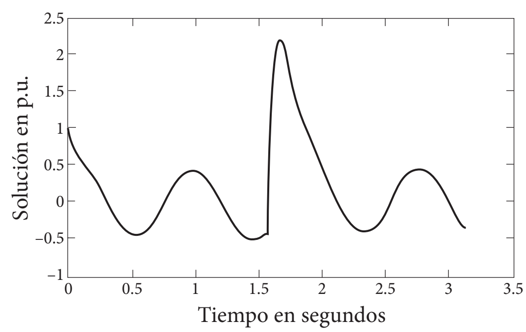




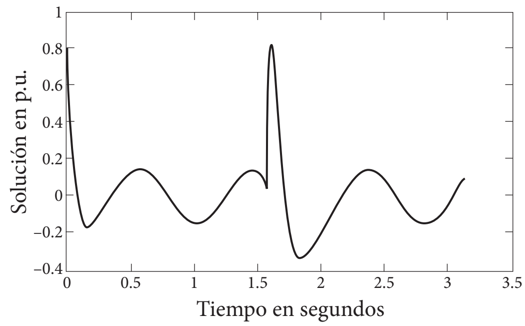
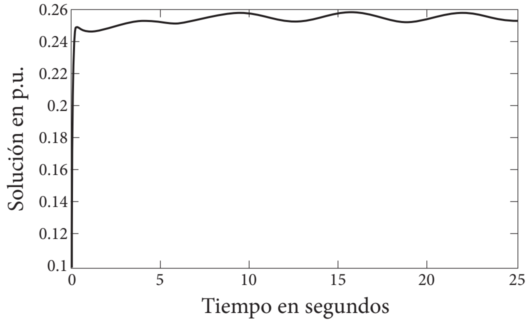
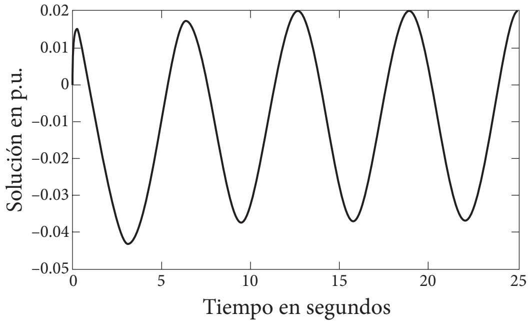
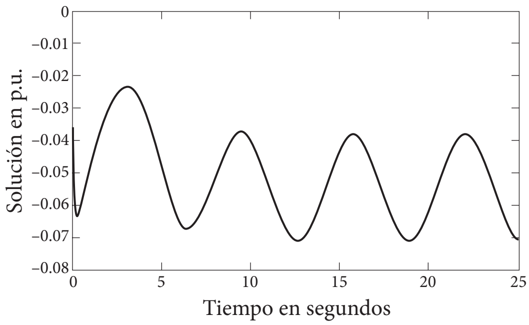
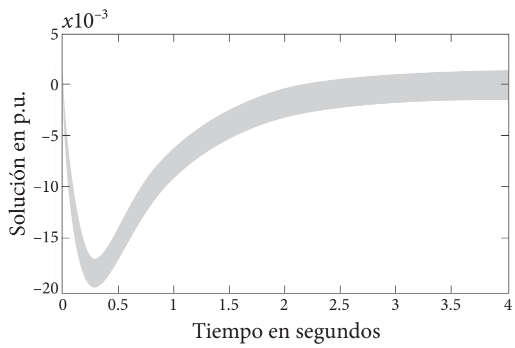


7.10.34 La gráfica de las soluciones es:

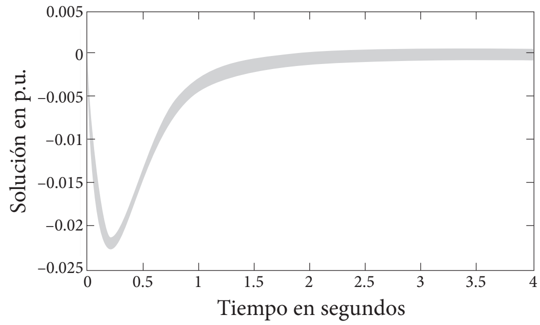


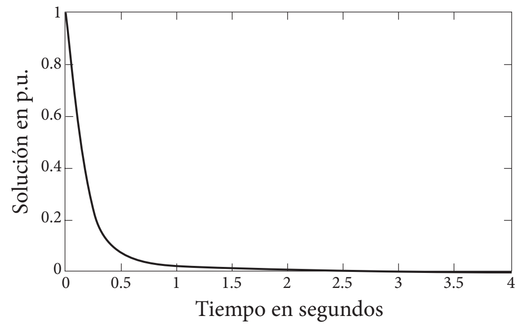


www.FreeLibros.me

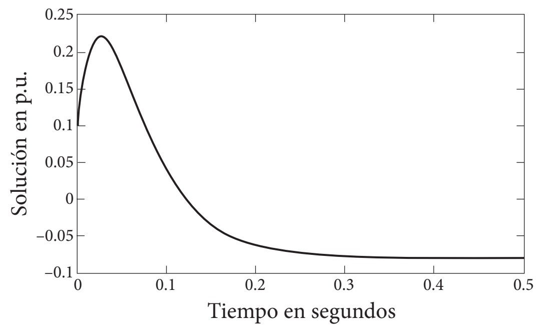
	
7.10.35 La gráfica de las soluciones es:	  
7.10.36 La gráfica de las soluciones es:	

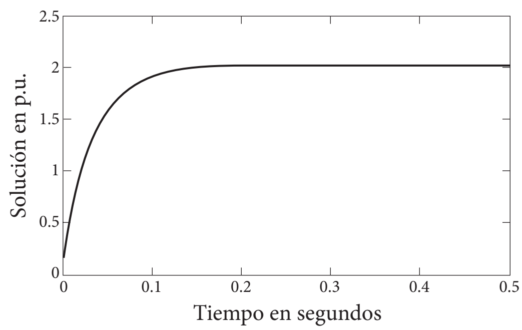
www.FreeLibros.me

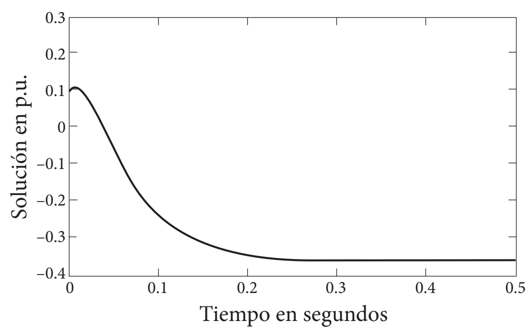




7.10.37 La gráfica de las soluciones es:

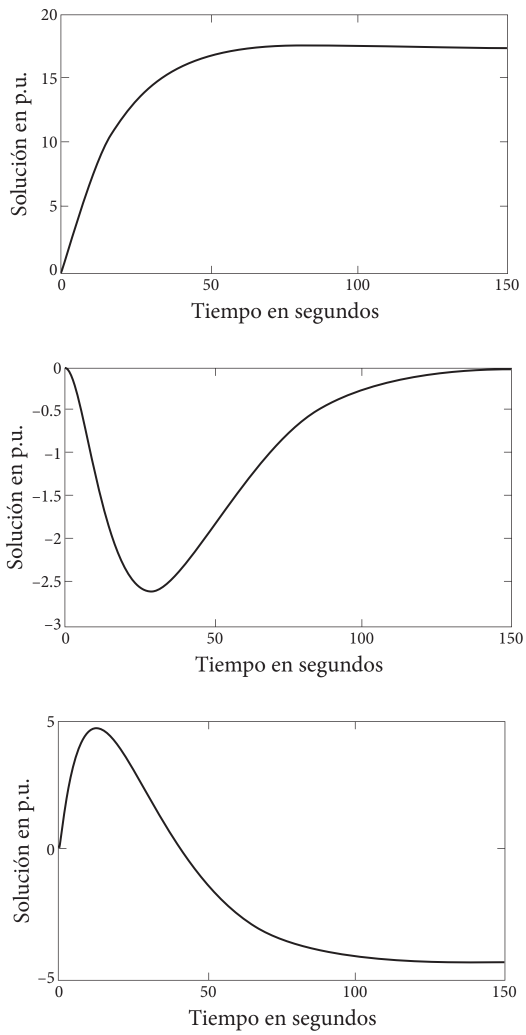






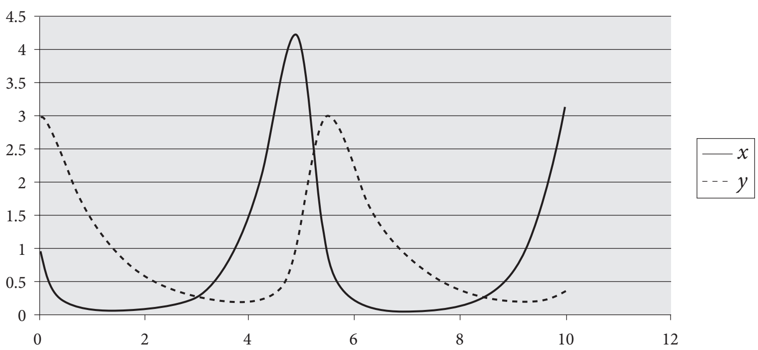
www.FreeLibros.me

7.10.38 La gráfica de las soluciones es:



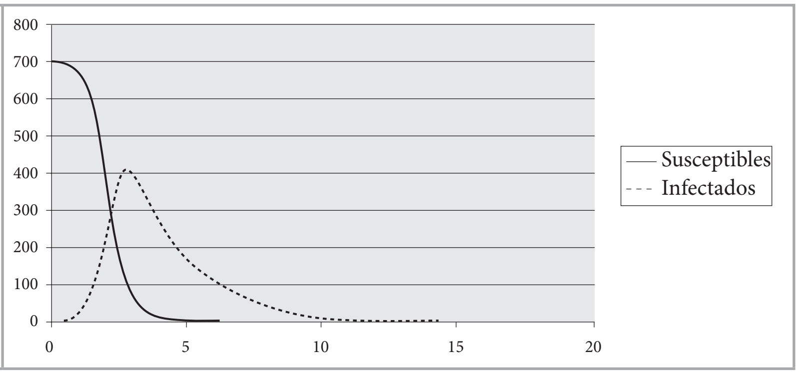
7.10.39 Usando un tamaño de paso $h = \frac{12}{512}$, se obtiene que $T(12) \approx 80.0000208$. La solución exacta al problema es $T(t) = 72\left(\frac{13}{18}\right)^{\frac{1}{12}t} + 28$, por lo que la solución exacta es $T(12) = 80$

7.10.40 La gráfica de las soluciones es:



www.FreeLibros.me

7.10.41 Usando el método de Runge-Kutta de cuarto orden para sistemas, con un tamaño de paso de 20/1 024, se obtienen las siguientes aproximaciones:



7.10.42 El sistema de cuatro ecuaciones resultante es

$$\frac{dx_1}{dt} = x_2$$

$$\frac{dx_2}{dt} = -g\frac{mx_1}{r^3}$$

$$\frac{dx_3}{dt} = x_4$$

$$\frac{dx_4}{dt} = -g\frac{mx_3}{r^3}$$

$$r = \sqrt{{x_1}^2 + {x_3}^2},\ x_1(0) = 1,\ x_2(0) = 0,\ x_3(0) = 0,\ x_4(0) = 1$$

7.10.43 Usando el método de Runge-Kutta de cuarto orden, se tiene que para $t \approx 69$ minutos, el tanque estará vacío.

Capítulo 8

8.10.1 $\mathbf{J} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$ $\quad \mathbf{V} = [\mathbf{V}_1 \quad \mathbf{V}_2 \quad \mathbf{V}_3 \quad \mathbf{V}_4 \quad \mathbf{V}_5] = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0.5 & 0.375 & 0.437 & 0.386 \\ 0 & 0 & 0.250 & 0.125 & 0.195 \\ 0 & 0 & 0 & 0.125 & 0.015 \\ 0 & 0 & 0 & 0 & 0.031 \end{bmatrix}.$

8.10.2 $\mathbf{J} = \begin{bmatrix} 2 & 1 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}$

8.10.3 $\mathbf{J} = \begin{bmatrix} 3 & 1 & 0 & 0 & 0 \\ 0 & 3 & 1 & 0 & 0 \\ 0 & 0 & 3 & 1 & 0 \\ 0 & 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix}$ $\quad \mathbf{V} = [\mathbf{V}_1 \quad \mathbf{V}_2 \quad \mathbf{V}_3 \quad \mathbf{V}_4 \quad \mathbf{V}_5] = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0.5 & 0.625 & 0.677 \\ 0 & 0 & 0.5 & 0.125 & 0.218 \\ 0 & 0 & 0 & 0.25 & 0.020 \\ 0 & 0 & 0 & 0 & 0.083 \end{bmatrix}$

www.FreeLibros.me

8.10.4 $\mathbf{J}=\begin{bmatrix}4&1&0&0&0\\0&4&1&0&0\\0&0&4&1&0\\0&0&0&4&1\\0&0&0&0&4\end{bmatrix}$ $\mathbf{V}=[\mathbf{V}_1\quad\mathbf{V}_2\quad\mathbf{V}_3\quad\mathbf{V}_4\quad\mathbf{V}_5]=\begin{bmatrix}1&1&1&1&1\\0&0.5&-0.25&1.375&-2.437\\0&0&0.5&-0.75&2.375\\0&0&0&0.5&-1.5\\0&0&0&0&0.25\end{bmatrix}$

8.10.5 $\mathbf{J}=\begin{bmatrix}5&1&0&0&0\\0&5&1&0&0\\0&0&5&1&0\\0&0&0&5&1\\0&0&0&0&5\end{bmatrix}$ $\mathbf{V}=[\mathbf{V}_1\quad\mathbf{V}_2\quad\mathbf{V}_3\quad\mathbf{V}_4\quad\mathbf{V}_5]=\begin{bmatrix}1&1&1&1&1\\0&1&-1&2&-1.5\\0&0&1&-1&2\\0&0&0&1&-1.5\\0&0&0&0&0.5\end{bmatrix}$

8.10.6 $\mathbf{J}=\begin{bmatrix}6&1&0&0&0\\0&6&1&0&0\\0&0&6&1&0\\0&0&0&6&1\\0&0&0&0&6\end{bmatrix}$ $\mathbf{V}=[\mathbf{V}_1\quad\mathbf{V}_2\quad\mathbf{V}_3\quad\mathbf{V}_4\quad\mathbf{V}_5]=\begin{bmatrix}1&1&1&1&1\\0&0.5&0.5&0.5&0.5\\0&0&0.5&0.5&0.5\\0&0&0&0.5&0.375\\0&0&0&0&0.125\end{bmatrix}$

8.10.7 $\mathbf{J}=\begin{bmatrix}7&1&0&0&0\\0&7&1&0&0\\0&0&7&1&0\\0&0&0&7&1\\0&0&0&0&7\end{bmatrix}$ $\mathbf{V}=[\mathbf{V}_1\quad\mathbf{V}_2\quad\mathbf{V}_3\quad\mathbf{V}_4\quad\mathbf{V}_5]=\begin{bmatrix}1&1&1&1&1\\0&1&0.5&0.916&0.527\\0&0&0.5&0.083&0.472\\0&0&0&0.166&-0.013\\0&0&0&0&0.041\end{bmatrix}$

8.10.8 $\mathbf{J}=\begin{bmatrix}8&1&0&0&0\\0&8&1&0&0\\0&0&8&1&0\\0&0&0&8&1\\0&0&0&0&8\end{bmatrix}$ $\mathbf{V}=[\mathbf{V}_1\quad\mathbf{V}_2\quad\mathbf{V}_3\quad\mathbf{V}_4\quad\mathbf{V}_5]=\begin{bmatrix}1&1&1&1&1\\0&0.5&0.5&0.5&0.5\\0&0&0.5&0.5&0.5\\0&0&0&0.25&0.25\\0&0&0&0&0.25\end{bmatrix}$

8.10.9 $\mathbf{J}=\begin{bmatrix}9&1&0&0&0\\0&9&0&0&0\\0&0&9&1&0\\0&0&0&9&0\\0&0&0&0&9\end{bmatrix}$ $\mathbf{V}=[\mathbf{V}_1\quad\mathbf{V}_2\quad\mathbf{V}_3\quad\mathbf{V}_4\quad\mathbf{V}_5]=\begin{bmatrix}1&1&0&1&0\\0&0&1&1&0\\0&0&0&1&1\\0&1&0&-1&0\\0&0&0&1&0\end{bmatrix}$

8.10.10 $\mathbf{T}=\begin{bmatrix}2.0000&-14.0712&0&0&0&0\\-14.0712&13.1263&-14.5760&0&0&0\\0&-14.5760&2.2747&-2.8965&0&0\\0&0&-2.8965&-1.9997&-5.1758&0\\0&0&0&-5.1758&-3.4884&-1.0504\\0&0&0&0&-1.0504&5.0871\end{bmatrix}$

www.FreeLibros.me

8.10.11
$$\mathbf{T}=\begin{bmatrix} 5.0000 & -13.8924 & 0 & 0 & 0 & 0 & 0 \\ -13.8924 & 28.2228 & 16.1908 & 0 & 0 & 0 & 0 \\ 0 & 16.1908 & 3.9660 & -2.4661 & 0 & 0 & 0 \\ 0 & 0 & -2.4661 & 3.9272 & 3.1382 & 0 & 0 \\ 0 & 0 & 0 & 3.1382 & 0.0296 & 2.4614 & 0 \\ 0 & 0 & 0 & 0 & 2.4614 & -2.6213 & -3.4168 \\ 0 & 0 & 0 & 0 & 0 & -3.4168 & 8.4757 \end{bmatrix}$$

8.10.12
$$\mathbf{T}=\begin{bmatrix} 2.0000 & -16.0000 & 0 & 0 & 0 & 0 & 0 & 0 \\ -16.0000 & 30.0313 & 15.5863 & 0 & 0 & 0 & 0 & 0 \\ 0 & 15.5863 & 6.2426 & 4.4650 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4.4650 & 0.2547 & 0.2746 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.2746 & 2.8871 & -6.7738 & 0 & 0 \\ 0 & 0 & 0 & 0 & -6.7738 & -2.3461 & 6.7154 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6.7154 & -0.4238 & -0.8686 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.8686 & -1.6457 \end{bmatrix}$$

8.10.13
$$\mathbf{T}=\begin{bmatrix} 5.00 & -17.37 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -17.37 & 27.97 & -15.33 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -15.33 & 4.60 & 7.46 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7.46 & -0.86 & -3.58 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -3.58 & -0.68 & 5.30 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5.30 & -0.51 & 2.92 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2.92 & 5.40 & 5.14 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5.14 & 4.00 & 4.65 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4.65 & 3.07 \end{bmatrix}$$

8.10.14
$$\mathbf{T}=\begin{bmatrix} 2.00 & -18.24 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -18.24 & 40.21 & 13.53 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 13.53 & 5.48 & -8.93 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -8.93 & -5.81 & -5.51 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -5.51 & -2.73 & 7.39 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7.39 & 1.46 & 3.43 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3.43 & -0.05 & -1.51 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1.51 & 7.89 & 0.72 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.72 & 0.77 & -2.43 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2.43 & -4.22 \end{bmatrix}$$

www.FreeLibros.me

8.10.15

$$
\mathbf{T} = \left[\begin{array}{ccccccccccc}
2.00 & -17.40 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-17.40 & 44.60 & -16.30 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -16.30 & 9.76 & 6.83 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 6.83 & -1.91 & -5.44 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -5.44 & -4.58 & -5.66 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -5.66 & -1.00 & 8.94 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 8.94 & -0.99 & -4.68 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -4.68 & -5.46 & 2.30 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.30 & -2.18 & -1.49 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.49 & 2.13 & -2.49 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2.49 & -7.34
\end{array}\right]
$$

8.10.16

$$
\mathbf{T} = \left[\begin{array}{ccccccccccccc}
5.0 & -21.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-21.3 & 47.4 & 26.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 26.2 & 5.5 & 5.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 5.8 & -1.3 & -8.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -8.1 & 3.0 & 4.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 4.8 & -2.6 & -6.8 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -6.8 & 2.0 & 5.7 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 5.7 & -7.8 & 4.6 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 4.6 & 0.9 & 4.4 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4.4 & -0.7 & 2.0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.0 & -3.4 & -1.7 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.7 & -3.5 & 2.4 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.4 & -1
\end{array}\right]
$$

8.10.17

$$
\mathbf{T} = \left[\begin{array}{cccccccccccccc}
1.0 & -20.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-20.8 & 55.6 & 24.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 24.1 & 4.8 & 8.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 8.0 & -1.8 & 5.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 5.8 & -1.3 & -7.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -7.0 & -2.8 & 6.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 6.1 & -3.9 & -6.8 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -6.8 & -0.8 & -8.2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -8.2 & -5.6 & -8.4 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -8.4 & -6.8 & -3.8 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3.8 & 0.2 & 4.2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4.2 & -5.8 & 4.5 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4.5 & -1.5 & -4.7 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -4.7 & -4.8
\end{array}\right]
$$

www.FreeLibros.me

8.10.18

$$
\mathbf{T}=\begin{bmatrix}
5.0 & -21.7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
-21.7 & 54.0 & -33.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & -33.0 & 9.1 & -6.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & -6.4 & -0.6 & -5.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & -5.1 & -8.5 & -7.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & -7.4 & -4.3 & -8.6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & -8.6 & 1.2 & -4.9 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & -4.9 & 0.9 & 5.6 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 5.6 & -2.6 & 5.2 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5.2 & -5.5 & -7.1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -7.1 & -8.0 & 5.5 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5.5 & 0.3 & 1.7 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.7 & -2.4 & -2.6 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2.6 & 2.8 & -3.6\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3.6 & -4.4
\end{bmatrix}
$$

8.10.19 $\text{vp}=11.5414 \quad \mathbf{V}=[0.4263 \quad 0.6570 \quad 0.6218]^T$

8.10.20 $\text{vp}=11.4568 \quad \mathbf{V}=[0.6206 \quad 0.3555 \quad 0.6989]^T$

8.10.21 $\text{vp}=11.9377 \quad \mathbf{V}=[0.6260 \quad 0.3903 \quad 0.6751]^T$

8.10.22 $\text{vp}=11.9142 \quad \mathbf{V}=[0.3291 \quad 0.4678 \quad 0.8203]^T$

8.10.23 $\text{vp}=14 \quad \mathbf{V}=[0.2661 \quad 0.5617 \quad 0.7834]^T$

8.10.24 $\text{vp}=10.5414 \quad \mathbf{V}=[0.8156 \quad 0.3740 \quad 0.4415]^T$

8.10.25 $\text{vp}=18.8037 \quad \mathbf{V}=[0.4407 \quad 0.4649 \quad 0.5694 \quad 0.5152]^T$

8.10.26 $\text{vp}=15.3353 \quad \mathbf{V}=[0.6363 \quad 0.3679 \quad 0.4912 \quad 0.4674]^T$

8.10.27 $\text{vp}=13.9397 \quad \mathbf{V}=[0.4324 \quad 0.4947 \quad 0.6283 \quad 0.4165]^T$

8.10.28 $$\Lambda=\begin{bmatrix}\lambda_1\\ \lambda_2\\ \lambda_3\end{bmatrix}=\begin{bmatrix}11.2579\\ 7.7992\\ 4.9429\end{bmatrix},\ \mathbf{V}=[\mathbf{V}_1 \quad \mathbf{V}_2 \quad \mathbf{V}_3]=\begin{bmatrix}0.5299 & 0.7485 & 0.0784\\ 0.5559 & -0.5831 & 0.5277\\ 0.6405 & -0.3157 & -0.8458\end{bmatrix}$$

8.10.29 $$\Lambda=\begin{bmatrix}\lambda_1\\ \lambda_2\\ \lambda_3\end{bmatrix}=\begin{bmatrix}13.6056\\ 6.3944\\ 4.0000\end{bmatrix},\ \mathbf{V}=[\mathbf{V}_1 \quad \mathbf{V}_2 \quad \mathbf{V}_3]=\begin{bmatrix}0.6775 & 0.8521 & 0.5570\\ 0.5201 & -0.3700 & -0.6583\\ 0.5201 & -0.3700 & 0.5064\end{bmatrix}$$

8.10.30 $$\Lambda=\begin{bmatrix}\lambda_1\\ \lambda_2\\ \lambda_3\end{bmatrix}=\begin{bmatrix}9.1004\\ 5.3389\\ 3.5607\end{bmatrix},\ \mathbf{V}=[\mathbf{V}_1 \quad \mathbf{V}_2 \quad \mathbf{V}_3]=\begin{bmatrix}0.4120 & 0.7092 & 0.1700\\ 0.5539 & 0.2055 & -0.8726\\ 0.7235 & -0.6744 & 0.4580\end{bmatrix}$$

8.10.31 $$\Lambda=\begin{bmatrix}\lambda_1\\ \lambda_2\\ \lambda_3\\ \lambda_4\end{bmatrix}=\begin{bmatrix}11.7528\\ 6.4664\\ 5.2266\\ 4.5543\end{bmatrix},\ \mathbf{V}=[\mathbf{V}_1 \quad \mathbf{V}_2 \quad \mathbf{V}_3 \quad \mathbf{V}_4]=\begin{bmatrix}0.4206 & 0.2362 & -0.0389 & 0.3314\\ 0.3491 & -0.0347 & 0.8909 & -0.5932\\ 0.4829 & 0.3974 & -0.2107 & -0.4121\\ 0.6841 & -0.8861 & -0.4005 & 0.6070\end{bmatrix}$$

www.FreeLibros.me

8.10.32 $\Lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{bmatrix} = \begin{bmatrix} 12.4224 \\ 7.2305 \\ 6.1411 \\ 4.2060 \end{bmatrix}, \quad \mathbf{V} = [\mathbf{V}_1 \quad \mathbf{V}_2 \quad \mathbf{V}_3 \quad \mathbf{V}_4] = \begin{bmatrix} 0.6282 & 0.7678 & -0.0247 & 0.0895 \\ 0.3812 & -0.3315 & 0.1738 & -0.7293 \\ 0.4326 & -0.4801 & 0.3261 & 0.1892 \\ 0.5224 & 0.2646 & -0.9289 & 0.6513 \end{bmatrix}$

8.10.33 $\Lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{bmatrix} = \begin{bmatrix} 12.9142 \\ 7.7564 \\ 6.3294 \\ 6.0000 \end{bmatrix}, \quad \mathbf{V} = [\mathbf{V}_1 \quad \mathbf{V}_2 \quad \mathbf{V}_3 \quad \mathbf{V}_4] = \begin{bmatrix} 0.4914 & 0.8641 & -0.3864 & 0.2582 \\ 0.5567 & -0.0317 & 0.6342 & -0.7746 \\ 0.4242 & -0.3337 & 0.4478 & -0.2582 \\ 0.5184 & -0.3756 & -0.4980 & 0.5164 \end{bmatrix}$

8.10.34 $\Lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{bmatrix} = \begin{bmatrix} 13.5566 \\ 6.6552 \\ 6.0000 \\ 4.7882 \end{bmatrix}, \quad \mathbf{V} = [\mathbf{V}_1 \quad \mathbf{V}_2 \quad \mathbf{V}_3 \quad \mathbf{V}_4] = \begin{bmatrix} 0.4026 & 0.0355 & 0.6742 & -0.3938 \\ 0.4152 & 0.8760 & -0.6742 & 0.6658 \\ 0.7096 & -0.4797 & -0.2697 & 0.4965 \\ 0.4026 & 0.0355 & -0.1348 & -0.3938 \end{bmatrix}$

8.10.35 $\Lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{bmatrix} = \begin{bmatrix} 12.0780 \\ 7.1969 \\ 6.4046 \\ 5.3205 \end{bmatrix}, \quad \mathbf{V} = [\mathbf{V}_1 \quad \mathbf{V}_2 \quad \mathbf{V}_3 \quad \mathbf{V}_4] = \begin{bmatrix} 0.4866 & 0.6626 & -0.2723 & -0.0833 \\ 0.4092 & -0.6153 & 0.9431 & 0.6624 \\ 0.5029 & -0.2043 & -0.0529 & -0.6734 \\ 0.5856 & -0.3751 & -0.1835 & 0.3176 \end{bmatrix}$

8.10.36 $\Lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{bmatrix} = \begin{bmatrix} 20.7326 \\ 3.5186 \\ -3.4709 \\ -2.7804 \end{bmatrix}, \quad \mathbf{V} = [\mathbf{V}_1 \quad \mathbf{V}_2 \quad \mathbf{V}_3 \quad \mathbf{V}_4] = \begin{bmatrix} 0.3939 & 0.8010 & -0.2555 & 0.3173 \\ 0.5476 & -0.5904 & -0.4815 & 0.7195 \\ 0.4718 & -0.0558 & 0.7588 & -0.6154 \\ 0.5678 & 0.0823 & -0.3564 & -0.0538 \end{bmatrix}$

www.FreeLibros.me

www.FreeLibros.me

Bibliografía

ALLEN, Myron B, *et al.*, 1998, *Numerical Analysis for Applied Science,* Nueva York, Wiley.

BURDEN, Richard L. *et al.*, 2002, *Análisis numérico*, México, Thomson Learning.

CHAPRA, Steven C. y Raymond P. Canele, 2007, *Métodos numéricos para ingenieros*, México, McGraw-Hill.

CHENEY, Ward y David Kincaid, 2008, *Numerical Mathematics and Computing*, Florence, Kentuky, Brooks/Cole Cengage Learning.

CORDERO, Alicia *et al.*, 2006, *Problemas resueltos de métodos numéricos*, Madrid, Thomson Learning.

GRAINGER, John J. *et al.*, 1996, *Análisis de sistemas de potencia*, México, McGraw-Hill.

GROSSMAN, Stanley I., 1996, *Álgebra lineal*, México, McGraw-Hill.

HSU, Hwei P., 1998, *Análisis de Fourier*, México, Addison-Wesley.

MARON, Melvin J. *et al.*, 1995, *Análisis numérico. Un enfoque practico*, México, CECSA.

MATHEWS, John H. *et al.*, 2000, *Métodos numéricos con Matlab*, Madrid, Pearson-Prentice Hall.

MATHEWS, John H. *et al.*, 1992, *Numerical Methods for Mathematics, Science, and Engineering*, Nueva Jersey, Prentice-Hall.

NAGLE, R. Kent *et al.*, 2005, *Ecuaciones diferenciales y problemas con valores en la frontera*, México, Pearson Educación.

NAKAMURA, Shoichiro, 1992, *Métodos numéricos aplicados con software*, México, Prentice-Hall.

NIEVES, Antonio y Federico C. Domínguez, 2002, *Métodos numéricos aplicados a la ingeniería*, México, CECSA.

RODRÍGUEZ, Francisco Javier, 2003, *Cálculo y métodos numéricos: teoría, algoritmos y problemas resueltos*, Madrid, Universidad Pontificia de Comillas.

STRANG, Gilbert, 1988, *Linear Algebra and its Applications*, Brooks/Cole Cengage Learning.

VAN VALKENBURG, 1997, *Análisis de redes*, México, Limusa.

WYLEY, C. Ray, 1982, *Matemáticas superiores para ingeniería*, México, McGraw-Hill

ZILL, Dennis G. y Michael R. Cullen, 2006, *Ecuaciones diferenciales con problemas de valores en la frontera*, México, Thomson Learning.

www.FreeLibros.me

www.FreeLibros.me

Índice analítico

A

B

www.FreeLibros.me

C

www.FreeLibros.me

www.FreeLibros.me

F

www.FreeLibros.me

G

H

www.FreeLibros.me

www.FreeLibros.me

www.FreeLibros.me

N

www.FreeLibros.me

O

P

www.FreeLibros.me

R

www.FreeLibros.me

S

www.FreeLibros.me

www.FreeLibros.me

W

www.FreeLibros.me

www.FreeLibros.me

www.FreeLibros.me

www.FreeLibros.me